Founded 1905

INTEGRATING PROCESS PLANNING AND SCHEDULING

BY

EXPLORING THE FLEXIBILITY OF PROCESS PLANNING

Wang Jiao

DEPARTMENT OF MECHANICAL ENGINEERING

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2003

ACKNOWLEDGEMENT

First of all, I wish to express my sincerely appreciation to my supervisors, Assoc. Prof. Zhang Yunfeng and Prof. Andrew Nee Yeh Ching, for their invaluable guidance, insightful comments, strong encouragement and personal concern both academically and otherwise throughout the course of the research.

I would like to thank the National University of Singapore for providing me with research scholarship to support my study.

Thanks are also given to my colleagues for their significant help and discussion: Miss Li Lin, Mr. Jia Hongzhong, Mr. Lin Qi and Ms. Zhang Liping. They have created a warm community in which we can enjoy our studies and lives in NUS.

I would also like to thank all my friends with whom I enjoyed my research and social life at NUS and all my well-wishers who have extended their support in one way or another.

Finally, my deepest thanks go to my parents, my sister and brother for their encouragement, moral support and love.

TABLE OF CONTENTS

List of Figures

List of Tables

List of Abbreviations

ATC	Apparent Tardiness Cost
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CAPP	Computer-Aided Process Planning
EDD	Earliest Due Date
GA	Genetic Algorithm
ICSS	Integrated CAPP-Scheduling System
IPPM	Integrated Process Planning Model
NLPP	Non-Linear Process Planning
OPM	Operation Method
OPT	Operation Type
PR	Precedence Relationship
SA	Simulated Annealing
SPT	Shortest Processing Time
TAD	Tool Access Direction

SUMMARY

This thesis presents a dynamic system for the integration of process planning and scheduling by exploring the flexibility of process planning in a batch-manufacturing environment. The integration is essential for the optimal use of production resources and generation of realistic process plans that can be readily executed with little or no modification. The integration is modeled in two levels, viz., process planning and scheduling, which are linked by an intelligent facilitator. The process planning module employs an optimization approach in which the entire plan solution space is first generated and a search algorithm is then used to find the optimal plan. Based on the result of scheduling, the performance measure information is presented to the user. The user then selects a particular performance measure to improve. Based on this requirement, the facilitator identifies a particular job and issues a change to its process plan solution space to obtain a satisfactory schedule through a progressive approach. Heuristic algorithms are developed and stored in the facilitator rule base for balancing machine utilization rate and reducing tardy jobs.

The uniqueness of this approach is characterized by the flexibility of the process planning strategy and the intelligent facilitator, which makes the full use of the plan solution space intuitively to reach a satisfactory schedule. The intelligent facilitator not only works as the interface to realize the communication between the

process planning module and the scheduling module, but also makes the three modules cooperate in a close-loop system, which can react dynamically to unsatisfactory qualities of scheduling results.

Chapter 1

INTRODUCTION

1.1 Background and Motivation

In the complex environment of a manufacturing system, the capability of producing an efficient production schedule is becoming a vital factor for a manufacturing business. Because of the inflexibility and deterministic approaches to decision making in a stochastic environment, and insufficient communication and exploitation of expertise, existing manufacturing systems cannot adequately meet the increasing requirements of production efficiency. In order to face new challenges, a shift of the manufacturing paradigm from the deterministic into new manufacturing prospect is needed. This research proposes and develops an innovative approach for the integration of process planning and scheduling activities to generate production schedules with high quality.

As commonly recognized, process planning and scheduling are the two main activities affecting the overall performance of a manufacturing system. Process planning translates the design data into a set of instructions to manufacture a part. Scheduling is an optimization process by which limited resources are allocated over time among parallel and sequential activities such that measures like tardiness and make-span are minimized.

Traditionally, process planning and scheduling are regarded as two separate tasks performed sequentially, and this may result in infeasible process plans or technologically non-optimal production schedules. Although computer-aided process planning (CAPP) has received great research effort in the past two decades (Alting and Zhang, 1989) (Elmaraghy, et al., 1993), it only emphasizes the technological requirements of a task, while scheduling involves the timing aspects of it. Generally speaking, process planning is in conflict with scheduling. Since process planning has neither a view nor control of the actual status of the production facilities, it might unnecessarily constrain scheduling if it blindly assigns manufacturing resources. Changes that occur during the implementation of a process plan are usually not fed back to the process planning function. Even though process plans are ideal and appear to be locally optimal to the process planning activity, the plans are frequently not truly optimal if evaluated based on some scheduling criteria. Real manufacturing scheduling problems are also dynamic in nature (Graves and Stephen, 1981) (Hadavi, et al., 1992). The scheduling function, with limited interactive communications and collaboration with the process planning function, has difficulties in taking advantages of the process plans. The characteristics of traditional manufacturing are:

(1) Scheduling follows process planning.

(2) Process planners assume there are unlimited resources in the shop floor and repeatedly select desirable machines.

(3) Process planning focuses on the technological requirements of a task without considering the job shop information.

(4) Scheduling is restricted by fixed process plans, which cannot be altered.

(5) Even if the shop floor conditions are considered during the process planning stage, the time delay between the planning phase and plan execution phase sometimes leads to infeasible process plans.

(6) As the real production environment is very complex, neither the process plans nor the planned schedules are truly followed in the shop floor. Without the feedback from the shop floor, it becomes very difficult to measure the quality or value of a plan for future enhancement.

Because of the aforementioned problems, process plans may not be followed exactly in the shop floor, which leads to a huge waste of resource and time in real time manufacturing systems. To solve these problems and to achieve satisfactory schedules, the integration of process planning and scheduling becomes essential. Thus, adopting the idea of integrating process planning and scheduling to improve schedule quality has been a research direction for intelligent manufacturing systems.

At the National University of Singapore, a process planning module has been developed (Ma, 1999) (Li, 2002). An integration algorithm for process planning and scheduling has also been developed (Saravanran, 2001), which focused on the performance improvement of machine utilization rate. In this thesis, the presented work focuses on developing an effective method for minimizing job tardiness and the implementation of the overall integration system.

1.2 Research Objectives

The main objective of this research is to develop an integration system for the process planning and scheduling activities for a batch-manufacturing environment. In order to achieve this objective, the following sub-objectives must be accomplished:

- The complexity of process plan optimization must be studied

- Development of a heuristic scheduling module that generates the schedule for job orders
- Development of a facilitator module that implements the integration of process planning and scheduling
- Development of heuristic rules for improving the schedule performance, including machine utilization rate and job tardiness
- Study on finding efficient modification algorithm for improving schedule quality performance

1.3 Organization of the Thesis

This thesis is organized into eight chapters:

In Chapter 2, a brief review of related works in the integration of process planning and scheduling are presented. In addition, the approaches for improving schedule quality by exploring scheduling strategies are introduced as well.

In Chapter 3, a description of system architecture integration is given.

In Chapter 4, the functions of the process planning module and scheduling module of the proposed integration system are described.

In Chapter 5, the facilitator module is described in detail. The development of this module is discussed focusing on the different functions of the module, which plays a pivotal role in the integration of the two functions—process planning and scheduling.

In Chapter 6, the implementation of the proposed integration system is given, followed by the description of the modules in the framework, viz., process planning, scheduling, and facilitator modules.

In Chapter 7, two case studies are given to illustrate the capabilities and advantages of the proposed integration system.

Finally, conclusions are stated, and recommendations for future work are presented in Chapter 8.

Chapter 2

LITERATURE REVIEW

The integration of process planning and scheduling activities has attracted great research interests in the past decade. Different researchers have proposed several integration approaches. Meanwhile, many researchers have been working on new scheduling strategies that produce schedules with high quality, such as minimized job tardiness. In this section, some of the approaches in the literature related to the research work of integrating process planning and scheduling and some research work on advanced scheduling functions are described.

2.1 Trends of Manufacturing Activities - Integration

Modern manufacturing environments are very much dynamic and unpredictable. The research and development in manufacturing activities has resulted in enormous improvements in product quality, efficiency and productivity. However, the isolated automation of different departments makes the inability of various units to generate the necessary information quickly, adequately and accurately. For top manufacturing companies, enterprise resource planning systems play a critical role in improving outdated infrastructures, gaining tighter control over internal operations, and driving down costs. To improve production efficiency, the need for greater integration of manufacturing activities arises. The techniques of an integrated intelligent system will speed up the process and improve the production efficiency, product quality and company competition (Currie and Tate, 1991). Implementing function integrations, such as the integration of process planning with product design (Bedworth et al., 1991) and the integration of process planning and scheduling, can make the manufacturing process have a better connection with customers and business partners, and to further boost the quality of production processes and reduce costs.

2.2 Integration of Process Planning and Scheduling

Automated process planning and scheduling have been receiving noteworthy attention from the research community since they are two of the major activities in a manufacturing system. Computer-aided process planning (CAPP) systems, developed in the past two decades or so, were intended to bridge the gap between computer-aided manufacturing (CAM) and computer-aided design (CAD), and to provide fast feedback to designers regarding detailed manufacturing information. A process plan specifies what raw materials are needed to produce a product, and what processes and operations are necessary to transform those raw materials into the final product. The outcome of process planning is the information for manufacturing processes and their parameters, and the identification of the machines, tools, and fixtures required to perform those processes.

Scheduling is another manufacturing system function that attempts to assign manufacturing resources to the processes indicated in the process plans in such a way that some relevant criteria, such as *due date* and *make-span* are met. Although there is a strong relation between process planning and scheduling, conventionally the two functions have been studied independently. As a common practice, process planning and scheduling tasks are performed separately.

Many problems may arise with the manufacturing system where process planning and scheduling are performed separately. Process planners usually assume that the shop is idle and that there are unlimited resources in the shop, and repeatedly select desirable machines. Thus when a process plan is going to be carried out, some constraints (such as limited resources or non-availability of machines) will be encountered, making the generated 'optimal' process plan infeasible or sub-optimal. Even if the dynamic shop status is considered, time delay between the planning phase and the plan execution phase may cause some troubles. Owing to the dynamic nature of a production environment, it is likely that by the time a part is ready to be manufactured, constraints that were used in generating the process plans may already have been changed to some degree, and thus the process plan has become sub-optimal or even totally invalid. Owing to the complexity of the real production environment, neither the process plans nor the planning schedules are truly followed in the shop. Without the feedback from the shop, it is difficult to measure the quality or effectiveness of a plan for future enhancement. To eliminate the problems mentioned above, the integration of process planning and scheduling has become essential and attracted great research interests in the past decade.

Over the last decade, there have been numerous research efforts towards the integration of process planning and scheduling (Tan and Khoshnevis, 2000). In general, the reported methods emphasize on two different approaches. The first one is based on the idea of using the dynamic just-in-time information of the job shop as input for generating process plans for incoming jobs. Such process plans are expected to be implemented with little or no modification. The second approach is based on the idea of exploring the alternative process plans for a given job in achieving a good schedule solution. This is a rather promising approach as it is designed towards achieving optimal process plans while satisfying the delivery requirements in the final schedule. Following this direction, the reported approaches, in general, can be further classified into two categories: the *iterative* approach and the *simultaneous* approach.

2.2.1 The *iterative* approach

Under this category, the CAPP system and the scheduling system are kept as two separate functional modules. For a given set of jobs, multiple feasible process plans are generated for each job. A top-prioritised plan for each job is then chosen and input to the scheduling system for generating a schedule. If the generated schedule is not satisfactory, a job is chosen and its current plan is replaced by another alternative plan, and the scheduling system generates a new schedule using the new process plan. This iterative process continues until a satisfactory schedule is found or no further improvement can be made. The implementation of this approach is rather straightforward. However, the vast solution space of process planning requires a highly efficient search algorithm in order to make this approach effective. Currently, the limitation among the reported developed systems is the lack of intelligent search strategy for choosing an appropriate process plan for a given job, thus making the search rather like a trial-and-error process. Some of the important integration systems under this category are described in the following sections.

The concept of non-linear process planning (NLPP) (Tonshoff et al., 1989) (Detand et al., 1992) (Kruth and Detand, 1992) (Kempenaers et al., 1996) is a proper means to realize the integration between process planning and scheduling. As opposed to traditional (linear) process plans, a NLPP does not contain one fixed operation sequence, but a set of alternative machine routings in an AND/OR graph. NLPPs will grow during the lifetime of the product. Other interesting alternative routings can be added later on. Feedback information coming from the workshop concerning performed times enables validation and improvement of the NLPPs. For each new order, a non-linear process plan is generated, i.e. a set of alternative machine routings is determined. Petri-nets can be used to model and solve the operations selection and sequencing problem (Kiritsis et al., 1999). A load-oriented scheduling system selects one alternative from the NLPPs, namely the routing that fits in best with the ongoing production, according to certain criteria. The use of NLPP influences the workshop performance on two levels: improvement in reactivity on disturbances; increase in schedule performance.

Critical path analysis has also been used in the integration of process planning and shop floor scheduling in small batch part manufacturing (Zijm, 1995). The approach explores possibilities to cut manufacturing leadtimes and to improve delivery performance. Using a set of initial process plans, a resource decomposition procedure is exploited to determine schedules which minimize the maximum lateness. However, the critical path approach makes the system not adaptable to other objective functions (such as balancing machine utilization rate) without adding more solution algorithms.

2.2.2 The *simultaneous* approach

The simultaneous approach is based on the idea of finding a solution (process plans for all the jobs and a schedule) from the combined solution space of process planning and scheduling. The basic elements are features that form the parts in the given jobs.

The objective is to find a process plan for each feature and a sequence in which features pass between machines subject to the technological constraints and some optimisation criteria with respect to process planning and scheduling performance. The strength of this approach is that the integration problem is modelled in a truly integrated manner with the whole solution space available. However, with such a vast solution space, finding even a feasible solution in a reasonable amount of time can be difficult. Moreover, operation, instead of feature, should be used as the basic element in process planning due to the fact that the total number of operations is not fixed for a given part, e.g., centre-drill + drill + ream can be replaced by centre-drill + mill. On the other hand, a pre-selected sequence among operations may affect the validity of an operation alternative (Ma et al., 2000). These conditional constraints must be considered in the search for an optimal solution. Some approaches under this category are described in the following sections.

Khoshnevis and Chen (1990) proposed the concept of dynamic CAPP, which combines process planning and scheduling functions and generates less costly schedules based on alternative process plans provided by the process planning function. A priority dispatching method with concurrent assignment algorithm is developed, which uses a time window scheme to control the number of assignments at each stage. The use of time window, however, limits the optimization within the scope of the time window and it is difficult to determine the actual window size.

The integrated process planning model (IPPM) proposed by Zhang and Mallur (1993, 1994) used a decision matrix to represent the integration problem. A fuzzy set operation to select set-ups and machine tools is also introduced. The weakness of the decision matrix method is that it requires predetermination of the contributions to the criterion for any given pair of feature and machine. This type of data is very difficult to estimate without considering the interaction between features and method selections. In case the performance criterion is to minimize the number of tardy jobs, it is hard to see the contribution of favoring one feature-machine assignment over the others.

Huang et al. (1995) developed a progressive approach for the integration of process planning and scheduling to reduce the computational complexity of the integration problem. In this approach, the process planning and scheduling activities are divided into three phases: preplanning, pairing planning and final planning. In the preplanning phase, the interaction is at a global level. In the pairing planning, the interaction is at a machine group level. In the final planning phase, the interaction is at a detailed level. Each setup within the selected process plan will be assigned to a specific machine. The criterion is the shortest manufacturing lead-time criterion. However, the effect of decisions made at one level cannot be seen immediately until it is evaluated by another level. Even when both levels see no improvement can be made, it does not necessary mean that the whole system reaches its global optimal.

Palmer (1996) proposed a simulated annealing (SA) approach to the integrated production scheduling. SA is a kind of neighborhood search method. It shares certain desirable properties with genetic algorithms and Tabu search. SA operates directly on the performance measure to be optimized. Generality is one of the primary reasons for the use of SA for integrated planning and scheduling. It requires a means of generating new configurations with minor variations to an existing one. Three plan change operators are introduced: reverse the order of the two sequential operations on a machine; reverse the order of the two sequential operations within a job; change the method used to perform an operation. With SA, the trade-off between execution time and solution quality can be controlled to some degree. However, the SA method tends to provide quality solutions at the cost of execution time, it performs deep search in a space that is hopelessly large in most real time settings.

Online integration of a process planning module with production scheduling (Mamalis et al., 1996) used an information flow, designed as a relational data model, to maintain the interaction between the process planning and the production scheduling systems and provides the dynamic feedback to the process planner. In the integration system, the decision-making module concerns its ability to react to modifications of the initial production conditions and provide optimal scheduling decisions. Furthermore, the information module based on relational data models and a CAD interface is capable of maintaining the stand-alone operation and the interaction between the process planning and production scheduling modules, which is a fundamental step towards system integration.

2.3 Approaches for Reducing Job Tardiness

Manufacturing scheduling problems have been studied extensively and several books have been published on this subject, such as those by Muth and Thompson (1963), Artiba and Elmaghraby (1997), Tapan (1999) and so on. Meeting due date is a key factor in evaluating scheduling performance and the problem of reducing tardy jobs has been addressed by many researchers over the last decade. The general approach towards reducing tardy jobs is to make the scheduling system more efficient and effective. A number of attempts have been made by different researchers to try to reduce job tardiness by developing an effective scheduling strategy.

Vepsalainen and Morton (1987) developed an apparent tardiness cost (ATC) heuristic for scheduling a unit capacity machine by minimizing the sum of weighted tardiness as a performance measure. Anderson and Nyirenda (1990) employed several rules to minimize tardiness in a job shop. The first is the combination of the shortest processing time (SPT) rule and the critical ratio rule, and the second is a combination of the SPT rule and the slack per remaining work rule. Schutten and Leussink (1996) proposed a branch-and-bound algorithm to minimize the maximum lateness of any job. The algorithm exploits the fact that an optimal schedule is contained in a specific subset of all feasible schedules. James (1997) demonstrated using tabu search to solve the common due date early/tardy machine scheduling problem. Different forms of the Tabu search are tested, including one based on a sequence of jobs solution space and another based on an early/tardy solution space. Chen and Lin (1999) proposed a multi-factor priority rule to reduce total tardiness cost in manufacturing cell scheduling. In their research, a multi-factor priority rule is presented to improve Weighted COVER rule. The presented new rule combines job processing time, job routing, job due date, and job-dependent tardiness cost for the scheduling in a manufacturing cell. In addition, Eom et al. (2002) suggested a three-phase heuristic to minimize the sum of the weighed tardiness. In the first phase, jobs are listed by the earliest due dates and then divided into smaller job sets according to a decision parameter. In the second phase, the sequence of jobs is improved through the use of the Tabu search method. In the third phase, jobs are allocated to machines using a threshold value and a look-ahead parameter.

The previously developed approaches are mainly based on finding high-quality scheduling rules. Although scheduling performance has been improved in those approaches, the integration of process planning and scheduling for reducing tardy jobs has been neglected. In the proposed research work, focus is on the reduction of tardy jobs through the integration of CAPP and scheduling.

Many research works have been carried out in the past years to stress the importance of the integration of process planning. In this chapter, different approaches towards the integration of process planning have been reviewed and developed integration systems have been briefly described. The reported approaches, in general, can be further classified into two categories: the iterative approach and the simultaneous approach. The approaches to reduce job tardiness by exploring the scheduling functions have also been reviewed. In this thesis, the proposed integration methodology aims at achieving schedule of high quality with minimized tardiness by exploring the flexibility of process planning. The developed integration system is able to achieve satisfactory process plans and schedules in an effective and efficient manner.

Chapter 3

SYSTEM ARCHITECTURE

The importance of the integration of process planning and scheduling for a dynamic manufacturing environment has been described in the previous chapters. In this chapter, the system architecture of the Integrated CAPP-Scheduling System (ICSS) will be described.

3.1 The New Integration Approach

The new integration approach is based on the idea of improving schedule performance measures by exploring the flexibility of process planning. In this approach, process planning and scheduling are kept as two separate functions. Upon receiving a set of jobs, the process plans of all jobs are generated independently followed by running a scheduling algorithm. The performance measures of the generated schedule are presented. The integration starts when a performance measure that needs improvement is identified. A particular job is then identified and its process planning solution space is modified accordingly. Its process plan is re-generated and a new schedule is also generated. In this way, the targeted schedule performance measure is improved. This whole integration process is iterative in nature.

3.2 System Architecture

Based on the new approach, an integration system is developed, which is named as the Integrated CAPP-Scheduling System (ICSS). The system architecture is illustrated in Figure 3.1. The system is comprised of three modules: CAPP module, scheduling module and facilitator module. The functions of the three modules are briefly described here.

The process planning module is able to generate a set of machining operations, called a process plan, to reach a specified goal, with given constraints while optimizing some stated criteria. Before running the process planning module, manufacturing information of the job has to be automatically input into the database, which includes: the *type* & *id* of features as well as the shape parameters of the features; machine information; and tool information. Then the process plan solution space of each job is generated. It includes all the possible machines, tools, tool access directions for manufacturing a job and the precedence relationships between the processing operations. An optimized process plan is generated and output finally.

Scheduling is a process by which limited resources are allocated over time among parallel and sequential activities such that measures like tardiness, work-in-progress inventory, and make-span are minimized. The input to the scheduling module is the process plans of all the jobs to be scheduled. Heuristic rules are used for generating a schedule.

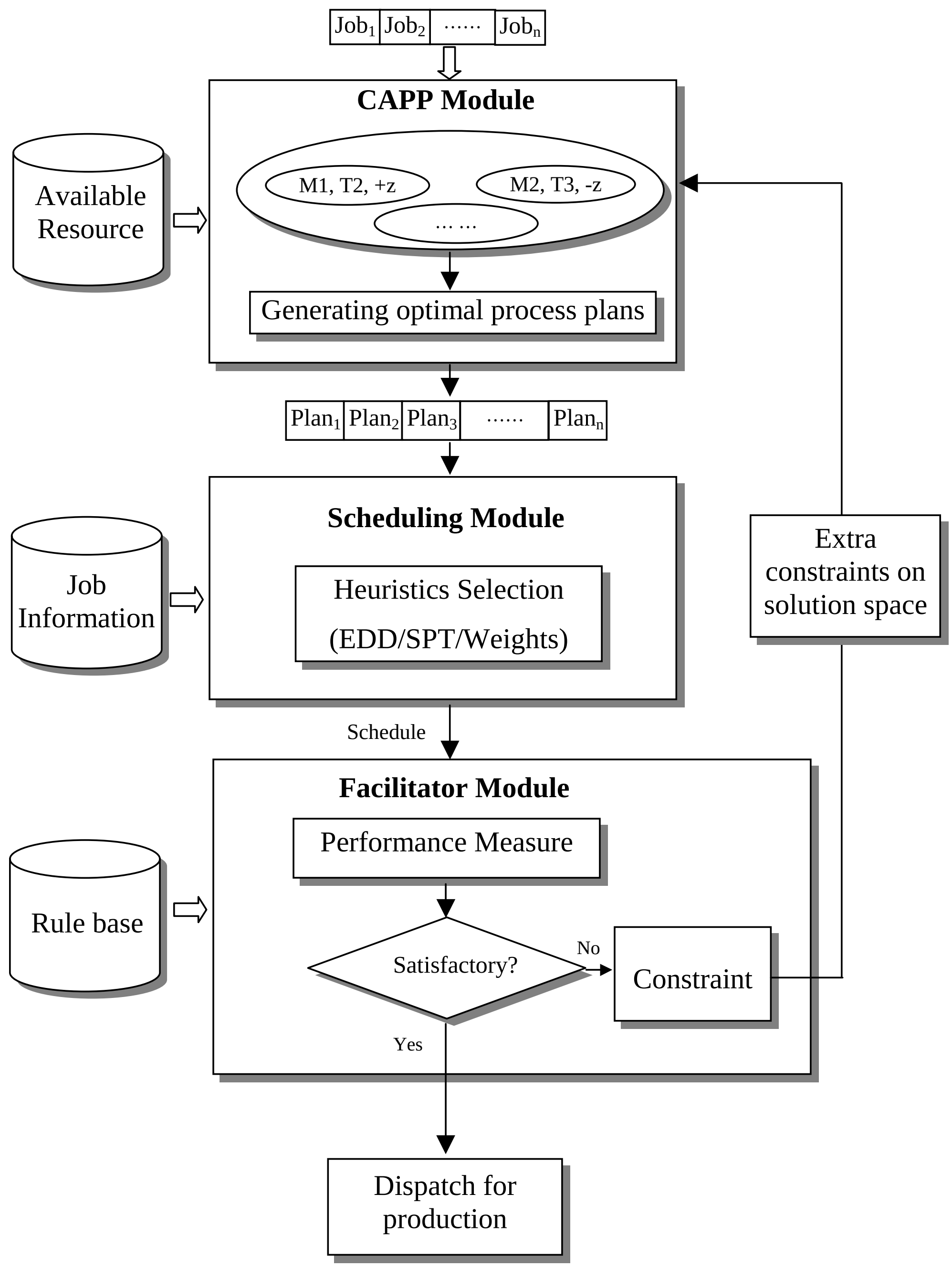


Figure 3.1 System architecture

The intelligent facilitator is incorporated as an integrator of process planning and scheduling. When a performance measure of the scheduling result is selected to be improved, a particular job will be identified for process plan solution space modification and regenerate the process plan. After that, a new schedule is generated. This process will be repeated until a satisfactory schedule is obtained. Thus the integration of process planning and scheduling is effected in a more dynamic way for a batch manufacturing environment.

Chapter 4

CAPP AND SCHEDULING MODULES

4.1 CAPP Module

Process planning is the generation of a set of machining operations, called a process plan, to reach a specified goal, with the given constraints while optimizing some stated criteria. A process plan describes the manufacturing processes for transforming a raw material to a completed part, within the available machine resources. Process planning can be regarded as a constrained optimization problem. Plans generated must meet various constraints imposed by the design specifications and the availability of manufacturing resources, and satisfy complex optimization criteria. Part feature is the most commonly used concept for part description in design, consequently a basic element for routing, sequencing and set-up planning.

In the proposed CAPP system (Li, 2002), the four steps to generate a process plan are: construct the process plan solution space, identify the precedence relationships (PRs) between operations, set up the objective function, and optimization. These steps are described as follows:

(1) *Construct the solution space*. The process plan solution space is composed of all feasible process plans. Generally, operations selection can be categorised into

two sub-stages: operation-type (OpT) selection and operation-method (OpM) selection. An OpT is an operation in name without being related to any machine (M), tool (T) and tool-approach-direction (TAD), e.g. drilling and end-milling. An OpM, in the form of M/T/TAD, indicates the M, T and TAD under which the OpT is to be executed. For each operation, the available machines and tools can be used for this operation and the tool access direction should all be identified and listed, which make up the solution space of the process plan.

(2) *Identify the precedence relationships (PRs) between operations*. For a given part, some machining operations should be performed before or after certain other operations. Precedence constraints will critically influence operations sequencing and set-up planning. Identifying all the precedence constraints is essential for solving the process plan optimization problem. Precedence relationships between operations are decided by fixture constraint, datum dependency and good machining practices.

(3) *Set up the objective function*. There are various cost functions to measure the effectiveness of a process plan. In this research, each of the two functions, i.e. minimizing total machining cost and minimizing total make-span, can be used as the criterion of optimization evaluation. The total production cost (*PC*) of a process plan can be calculated using the following equation:

$$PC = MC + TC + MCC + TCC + SCC \tag{5.1}$$

Where: *MC* – Machine cost index

TC – Tool cost index

MCC – Total machine change cost index

TCC – Total tool change cost index

SCC – Total set-up change cost index

The total processing time (*PT*) of a process plan can be calculated using the following equation:

$$PT = MT + MCT + TCT + SCT \tag{5.2}$$

Where: *PT* – Total processing time index

MT – Total machining time index

MCT – Total machine change time index

TCT – Total tool change time index

SCT – Total set-up change time index

Time and cost indices are used for calculating the processing time and cost, which are described in detail in (Li, 2002) and (Zhang, 1997) respectively.

(4) *Optimization*. Genetic Algorithm (GA) is used as the optimization search technique in the present system. GA performs searches based on the principle of natural selection and genetics. The unique characteristics of the GA, such as easy implementation and domain independence, make it more powerful than the conventional optimization methods for problems with large search space and the NP-hard problems (Zhang et al., 1997).

Figure 4.1 shows a sample part and all its features. A job shop containing 4 machines and 16 tools is considered. The machine and tool information is listed in Tables 4.1 and 4.2 respectively. The solution space of the sample part is shown in Table 4.3, in which the first column is the index of OpTs for processing the part and the second column is the index for the part features. It can be seen that a feature may need more than one operation. The third column listed all the possible OpMs for all the OpTs of the sample part.

Table 4.1 Machine database of the job shop

Machine Code	Machine Type	Table length (mm)	Table width (mm)	Travel X (mm)	Travel Y (mm)	Travel Z (mm)	Accuracy (mm)
M1	VERTICAL_MILLING	1300	280	850	400	400	0.02
M2	VERTICAL_CNC	1400	650	1200	600	700	0.01
M3	DRILLING	1000	280	850	400	400	0.1
M4	LATHING	1500	550	930	750	480	0.02

Table 4.2 Cutting tool database

Tool code	Tool type	Shank Dia (mm)	Dia (mm)	Flute Length (mm)	Whole Length (mm)	Angle
1	END_MILL	10.00	10.00	30.00	100.0	0
2	END_MILL	20.00	20.00	30.00	120.0	0
3	END_MILL	30.00	30.00	50.00	150.0	0
4	SIDE_MILL	40.00	100.0	10.00	10.00	0
5	DRILL	20.00	20.00	55.00	120.0	0
6	DRILL	30.00	30.00	50.00	100.0	0
7	DRILL	40.00	50.00	40.00	90.00	0
8	DRILL	50.00	70.00	100.0	200.0	0
9	CENTER_DRILL	20.00	5.00	20.00	70.00	0
10	ANGLE_CUTTER	45.00	50.00	20.00	50.00	4.5
11	DRILL	6.00	5.00	30.00	75.00	0
12	DRILL	8.00	8.00	30.00	80.00	0
13	DRILL	10.00	10.00	35.00	75.00	0
14	DRILL	15.00	50.00	50.00	75.00	0
15	LATHE	19.00	19.00	38.00	140.00	0
16	LATHE	25.00	25.00	38.00	160.00	0

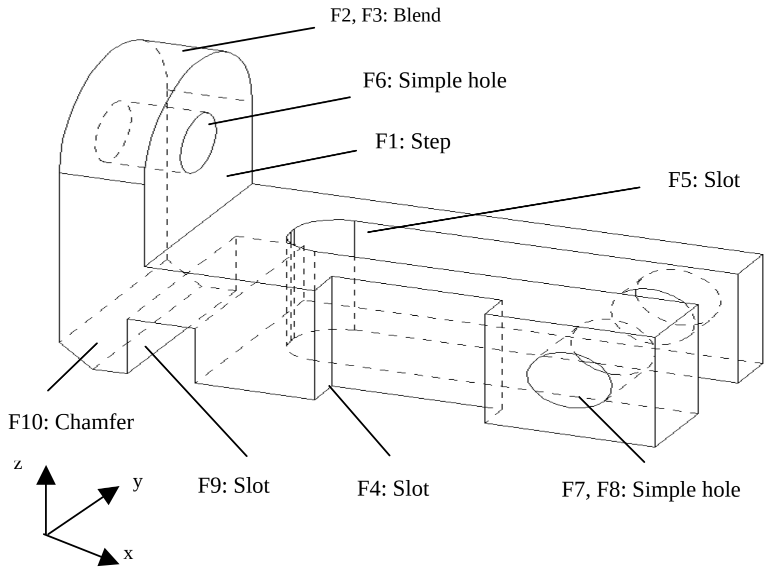


Figure 4.1 An example part with its features

Table 4.3 Process plan solution space

OpTs	Feature	Possible OpMs (M, T, TAD)
OpT1	F1: Step	(M1, T1, +x) (M1, T1, -x) (M1, T1, +y) (M1, T1, -z) (M1, T2, +x) (M1, T2, -x) (M1, T2, +y) (M1, T2, -z) (M1, T3, +x) (M1, T3, -x) (M1, T3, +y) (M1, T3, -z) (M1, T4, +x) (M1, T4, -x) (M1, T4, +y) (M1, T4, -z) (M2, T1, +x) (M2, T1, -x) (M2, T1, +y) (M2, T1, -z) (M2, T2, +x) (M2, T2, -x) (M2, T2, +y) (M2, T2, -z) (M2, T3, +x) (M2, T3, -x) (M2, T3, +y) (M2, T3, -z) (M2, T4, +x) (M2, T4, -x) (M2, T4, +y) (M2, T4, -z)
OpT2	F2: Blend	(M1, T1, -x) (M1, T1, +x) (M1, T1, -z) (M1, T2, -x) (M1, T2, +x) (M1, T2, -z) (M1, T3, -x) (M1, T3, +x) (M1, T3, -z) (M1, T4, -x) (M1, T4, +x) (M1, T4, -z) (M2, T1, -x) (M2, T1, +x) (M2, T1, -z) (M2, T2, -x) (M2, T2, +x) (M2, T2, -z) (M2, T3, -x) (M2, T3, +x) (M2, T3, -z) (M2, T4, -x) (M2, T4, +x) (M2, T4, -z)
OpT3	F3: Blend	(M1, T1, -x) (M1, T1, +x) (M1, T1, -z) (M1, T2, -x) (M1, T2, +x) (M1, T2, -z) (M1, T3, -x) (M1, T3, +x) (M1, T3, -z) (M1, T4, -x) (M1, T4, +x) (M1, T4, -z) (M2, T1, -x) (M2, T1, +x) (M2, T1, -z) (M2, T2, -x) (M2, T2, +x) (M2, T2, -z) (M2, T3, -x) (M2, T3, +x) (M2, T3, -z) (M2, T4, -x) (M2, T4, +x) (M2, T4, -z)
OpT4	F4: Slot	(M1,T1,+y) (M2,T1,+y) (M1,T3,+y) (M2,T3,+y)
OpT5	F5: Slot	(M1, T1, -z) (M1, T1, +z) (M1, T1, -x) (M1, T2, -z) (M1, T2, +z) (M1, T2, -x) (M1, T3, -z) (M1, T3, +z) (M1, T3, -x) (M2, T1, -z) (M2, T1, +z) (M2, T1, -x) (M2, T2, -z) (M2, T2, +z) (M2, T2, -x) (M2, T3, -z) (M2, T3, +z) (M2, T3, -x)
OpT6	F6: Hole	(M1,T9,-x) (M1,T9,+x) (M2,T9,-x) (M2,T9,+x) (M3,T9,-x) (M3,T9,+x)
OpT7		(M1,T6,-x) (M1,T6,+x) (M2,T6,-x) (M2,T6,+x) (M3,T6,-x) (M3,T6,+x)
OpT8	F7: Hole	(M1,T9,+y) (M2,T9,+y) (M3,T9,+y)
OpT9		(M1,T14,+y) (M2,T14,+y) (M3,T14,+y)
OpT10	F8: Hole	(M1,T9,-y) (M2,T9,-y) (M3,T9,-y)
OpT11		(M1,T14,-y) (M2,T14,-y) (M3,T14,-y)
OpT12	F9: Slot	(M1,T1,+z) (M2,T1,+z)
OpT13	F10: Chamfer	(M1, T1, -y) (M1, T1, +y) (M1, T2, -y) (M1, T2, +y) (M1, T3, -y) (M1, T3, +y) (M1, T4, -y) (M1, T4, +y) (M2, T1, -y) (M2, T1, +y) (M2, T2, -y) (M2, T2, +y) (M2, T3, -y) (M2, T3, +y) (M2, T4, -y) (M2, T4, +y)

After the process planning module runs the GA algorithm, the optimal process plan is generated, which is shown in Table 4.4. The evolution of production cost is shown in Figure 4.2, in which the minimized production cost is reached after 43 generations.

Table 4.4 The process plan of the sample part

Order	1	2	3	4	5	6	7	8	9	10	11	12	13
Op-id	12	1	5	2	3	6	8	9	7	13	4	10	11
M-id	01	01	01	01	01	01	01	01	01	01	01	01	01
T-id	01	01	01	01	01	09	06	09	14	09	14	01	01
TAD	+z	-x	-x	-x	-x	-x	+y	+y	+y	+y	+y	-y	-y
The total production cost is 675; Total production time is 473.													

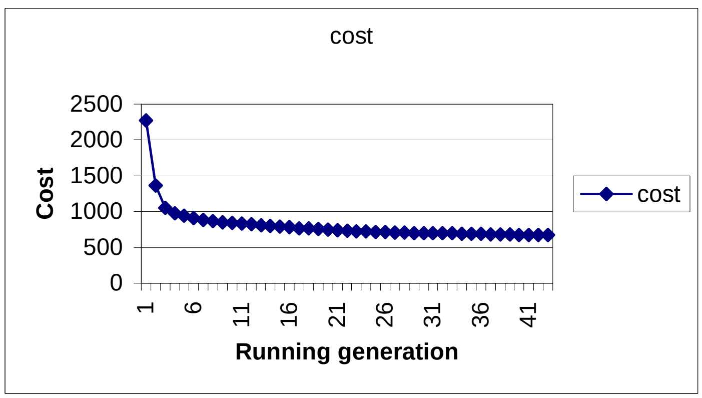


Figure 4.2 The evolution of production cost

4.2 Scheduling Module

Scheduling is a process by which limited resources are allocated over time among parallel and sequential activities such that measures like tardiness, work-in-progress inventory, and make-span are minimized. Much effort has been made in developing an efficient scheduling system. In the development of the scheduling systems, the following assumptions are frequently made:

a) Each machine can process only one job at any one time;

b) Each job is processed on one machine at any one time;

c) The operation cannot be interrupted;

d) The release time of jobs is usually ignored, which means all jobs are available at the commencement of processing;

e) Any time required to adjust or setup is usually ignored or included in the processing time; and

f) The processing time and technological constraints are deterministic and known in advance.

In the present system, a heuristic scheduling system (Figure 4.3) was developed for the generation of schedules of a set of jobs. This was developed based on the critical job procedure in which the first job in the queue is scheduled first throughout the job shop before proceeding to the next job in the queue. This scheduling system provides three optional heuristics (Baker, 1974; Morton and Pentico, 1993): earliest due date (EDD), shortest processing time (SPT), or job weightage (weights).

(1) *Weights:* The highest priority is given to the job with the highest weight. The priority of job assignment decreases with decreasing weights (w_j).

(2) *Earliest Due Date (EDD)*: The highest priority is given to the job with the earliest due date. The priority of job assignment decreases with increasing due date (d_j).

(3) *Shortest Processing Time (SPT):* The highest priority is given to the job with the shortest processing time. The priority of job assignment decreases with increasing total processing time (p_j).

where, j - job number

w_j - weight of job j

d_j - due date of job j

p_j - processing time of job j. It is the sum of processing times of all its operations.

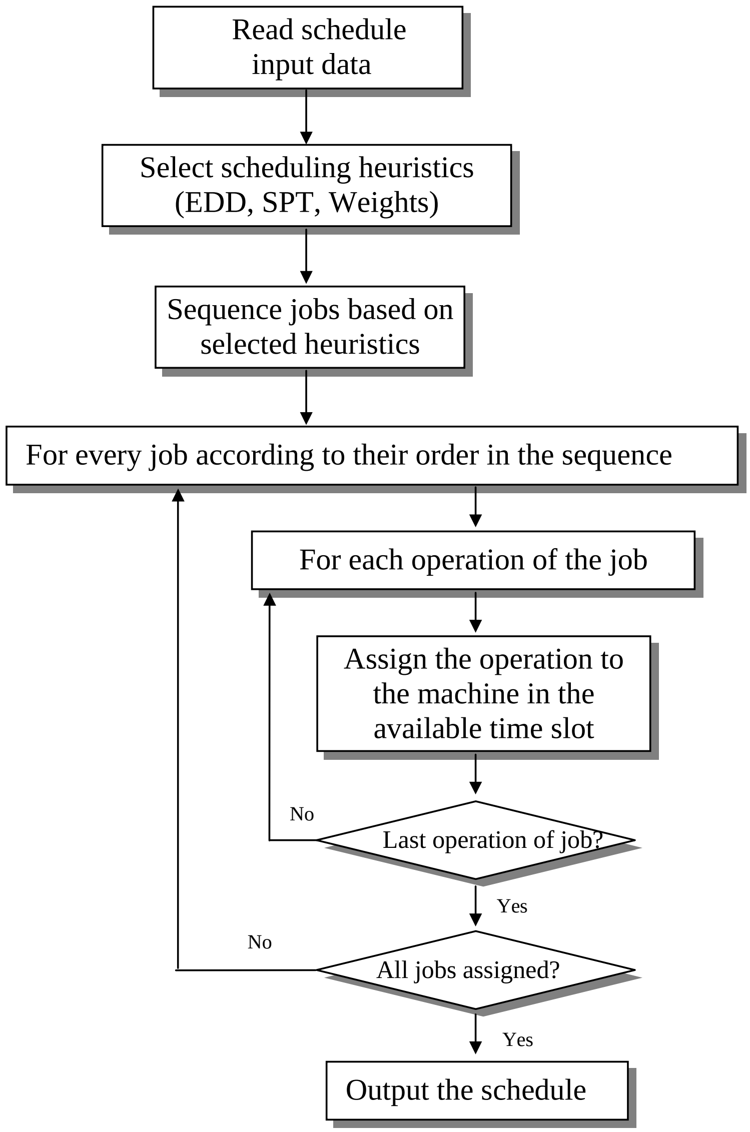


Figure 4.3 Flow chart of the scheduling system

It was noted that many optimization approaches for scheduling have been developed over the years (Tan and Khoshnevis, 2000). This simple heuristic-based approach was chosen mainly due to that the focus of this work is on exploring the flexibility of process planning for the integration with scheduling. This selection, however, does not limit the use of more advanced scheduling algorithms for this integration approach.

Chapter 5

THE FACILITATOR FOR INTEGRATION

The facilitator module is incorporated as an integrator of process planning and scheduling. Process planning concerns itself with technological requirements for manufacturing a part whereas scheduling deals with timing and resources allocation aspects. The facilitator module described in this chapter is developed in such a way that it exchanges the necessary information, in the form of feedback, between the two functions and helps to attain a better overall performance.

5.1 Facilitator Functions

The facilitator module (Figure 5.1) helps to achieve the integration by providing feedback to the process planning module in the form of constraints that the process plan has to follow. Upon receiving a set of jobs, the process plans of all jobs are generated independently followed by running a scheduling algorithm. The performance measures of the generated schedule are presented. If a performance measure is identified to be improved, the facilitator will generate constraints based on the performance measure and modify the process planning solution space by applying the constraints. Then the process plan is re-generated and a new schedule is also generated. In this way, the targeted schedule performance measure is improved. This process continues until a satisfactory schedule is achieved or no further improvement can be made.

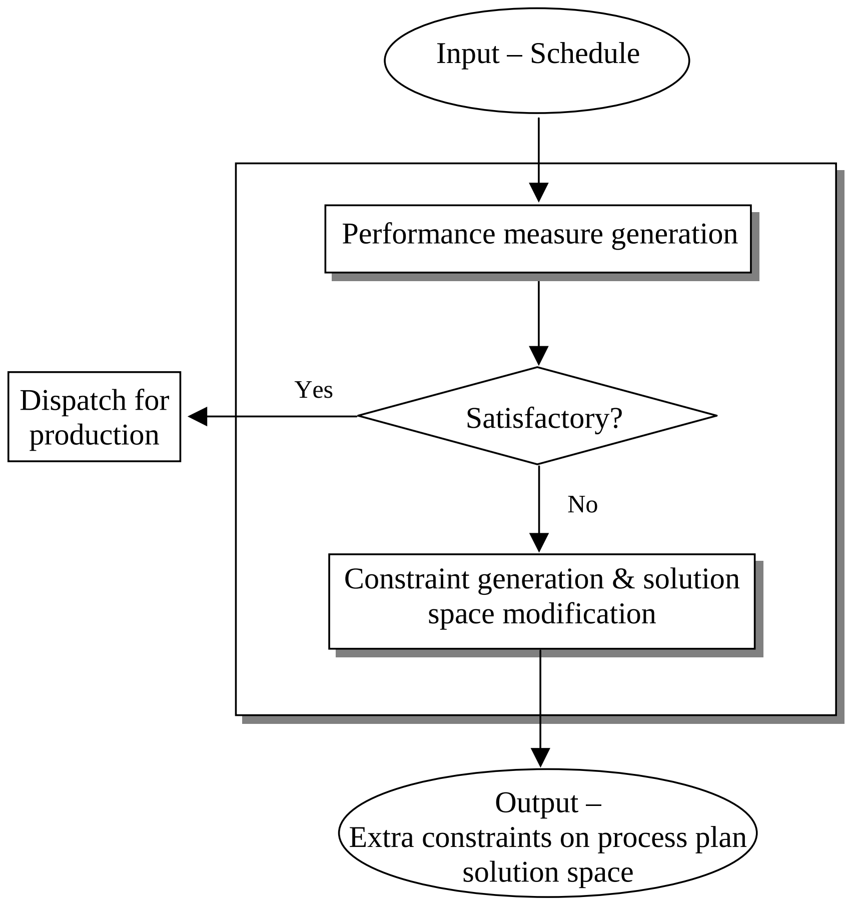


Figure 5.1 Facilitator functions

The algorithm of the facilitator module can be generally described as follows:

(1) Performance measure and evaluation. Generate the schedule performance measures (machine utilization and job tardiness) and display it graphically on the screen;

(2) Prompt the user to select a performance measure that needs improvement.

If all the performance measures are satisfactory

Dispatch the schedule to shop floor for production. Stop.

Else

Go on to step (3);

End

(3) Constraint generation and messaging, and regenerate process-planning solution space.

The details of this algorithm are given in the following sections.

5.2 Performance Measure Evaluation

The facilitator module starts by generating the performance measures of a generated schedule currently available. The performance measures include job tardiness and machine utilization rate.

5.2.1 Job tardiness

Lateness is the amount of time by which the completion time of a job exceeds its due date, i.e.,

$$L_j = C_j - d_j \tag{5.1}$$

where,

L_j - lateness of job 'j'

C_j - completion time of job 'j'

d_j - due date of job 'j'

Lateness measures the conformity of the schedule to a given due date. Negative lateness represents better service than requested while positive lateness represents poorer service.

Tardiness is defined as the lateness of a job if it fails to meet its due date, or zero otherwise (Baker, 1974),

$$T_j = \max\{0, L_j\} \tag{5.2}$$

where,

T_j - tardiness of job '*j*'

L_j - lateness of job '*j*'

The tardiness identifies the jobs which are completed beyond their stipulated due date and also gives the lateness of each job. Here, the ideal situation is that all the jobs are completed by their respective due dates.

5.2.2 Machine utilization rate

Machine utilization provides the loading of different machines in the job shop during the scheduling period. Machine utilization may be defined as (Palmer, 1996):

$$U_i = \frac{\sum_{j=1}^{n} p_{ij}}{C_{max} - a_i} \times 100\% \tag{5.3}$$

where,

U_i - the utilization of machine '*i*'

P_{ij} - the processing time of job '*j*' on machine '*i*'

a_i - the initial availability date (scheduled start date)

C_{max} - the make-span (maximum completion time of all the jobs)

n - the number of jobs

Machine utilization rate is the given proportion of time a machine is active, between the start of its availability and the finish of the last operation on all machines. It helps to identify the over-utilized and under-utilized machines in the shop. Over-utilization hints potential breakdown of the machine and under-utilization hints the availability of the machine for more jobs.

5.3 Heuristics for Constraint Generation

Once the machine utilization rate and job tardiness are displayed on the screen, the user may request the system to improve the schedule by the following two ways:

(1) Reduce the total number of tardy jobs;

(2) Select a machine to reduce its utilization rate.

The system then needs to select a job and change its process plan solution space. This is done based on heuristics. These heuristics are the key to integration. In the following sections, various heuristics for the two different requirements are described.

5.3.1 One basic term

One basic term used in developing heuristics for constraints generation is the *operation waiting time (OpWT),* which corresponds to the time period of one operation of a job in which it is waiting to be processed by a machine, which is busy during that time period. For a set of jobs, after the process plan of each job is generated and the schedule is produced, the operation waiting time can be calculated as:

$$OpWT(j, k) = StartT(j, k) - EndT(j, k-1) \qquad (5.4)$$

Where,

$OpWT(j, k)$ – The operation waiting time of the 'k'th operation of Job'j'

StartT(j, k) – The starting time of the 'k'th operation of Job'j'

EndT(j, k-1) – The ending time of the 'k-1'th operation of Job'j'

Since the operation waiting time is an essential factor in deciding manufacturing efficiency and is frequently used during the evaluation and modification process, one dynamic data-recording table is maintained in the system database.

5.3.2 Heuristics for reducing tardy jobs

Considered as the key factor in deciding the timing aspect of a job, reducing operation waiting time is the general objective of the proposed modification strategy. The general procedures of constraint generation process for reducing tardy jobs are shown in Figure 5.2. Supposing an operation of a tardy job has a non-zero waiting time to a machine, by selecting a machine that is idle at that moment will possibly remove this waiting time, which may in turn reduce the tardiness of the job.

Based on the aforementioned strategy, four heuristic rules towards different types of scenarios are developed for reducing tardy jobs. In the performance evaluation step, the tardy jobs are identified, which is the input of the tardy job modification algorithm. The general job modification heuristic is summarized below.

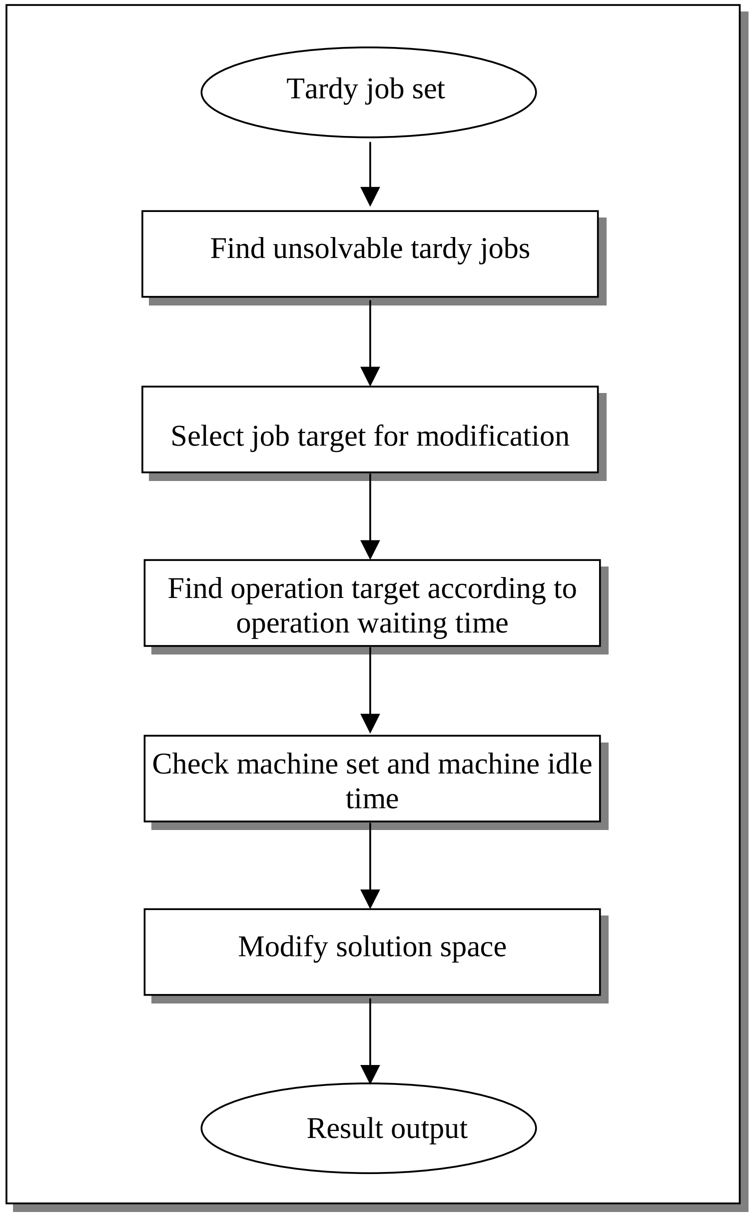


Figure 5.2 General constraint generation procedures

Begin

(a) Find unsolvable tardy jobs

For each tardy job, check whether the job has the possibility to meet the due date by comparing job processing time and the maximum allowed time. The maximum allowed time of a job is the interval between the ready time and due date. If the job's processing time is longer than its maximum allowed time, then the job cannot be delivered on time, and is consequently output as an unsolvable tardy job and released from the tardy job set.

(b) Select job target

Sort the tardy jobs and represent them as $\{J_{tdy-1}, J_{tdy-2}, \ldots, J_{tdy-n}\}$ *in ascending order of tardiness. Select the first job in the list and assign it to TarJ, i.e., TarJ =* J_{tdy-1}*.*

(c) Find operation

Check the schedule of the operations for TarJ. Find out the operations with non-zero operation waiting time. Set the operation with the longest waiting time as OpT_{tdy} *and the machine used in this operation is represented as* M_u*.*

(d) Check machine set

Check the process plan solution space of TarJ and find the OpMs of OpTarJ, the machine set of the OpMs is represented as $\{M_1, M_2, \ldots, M_m\}$*. If the machine set only has one component, i.e.,* M_u*, then set TarJ =* J_{tdy-2}*. Go to Step(c).*

Else go to (e).

(e) Solution space modification

Change the process plan solution space for OpT_{tdy} *according to a specific heuristic rule. There are totally four rules, which are described in the next section*

(f) Output result

Output the modification TarJ and its modified solution space.

End

The above heuristic describes one round of modification process. If the number of tardy jobs does not decrease in the resulting schedule or a new tardy job is generated, more iterations should be tried. Reducing the number of tardy jobs is a progressive approach and the process can be repeated until no further reduction of tardy job can be made.

Based on the aforementioned general heuristic rule, four modification heuristics have been developed for reducing tardy jobs:

- Cost-based Fine-tuning Rule (CFR),
- Cost-based Quick-tuning Rule (CQR),
- Time-based Fine-tuning Rule (TFR),
- Time-based Quick-tuning Rule (TQR).

In the process planning module, each of the two objective functions, i.e., minimizing total machining cost and minimizing total make-span, can be used as the process plan optimization target. Whether cost or time is set as the optimization target decides whether a cost-based rule or a time-based rule is selected.

In the process planning stage, if cost is the optimization target, low-cost machine (but normally slow) is preferred and frequently selected in generating an optimal process plan. This will usually cause jobs waiting to be processed on the low-cost machine and the higher-cost machine idle in the resulted schedule. In this case, cost-based heuristic rule CFR is selected for solution space modification, which is summarized below (the steps that are the same as that of the general heuristic are not repeated). In CFR, the solution space of one operation of one tardy job is modified each time, which makes the modification iterations a fine-tuning process. This could effectively, to a large extent, prevent the scenario in which the improvement on one performance measure results the worsening of other performance measures. The CFR is given below:

Cost-based Fine-tuning Rule (CFR)

Begin

(a) ……

(b) ……

(c) ……

(d) ……

(e) Solution space modification

Remove M_u from the machine set $\{M_1, M_2, \ldots, M_m\}$.

(f) ……

End

Besides CFR, a quick-tuning rule CQR is also provided, which is a faster way of modification and makes a larger change to the solution space in one round of modification compared with that of CFR. In each round of the solution space modification, one operation method of each tardy job will be modified. This can make the progressive modification need less iteration and consequently speed up the process. The details of CQR are described below. Although CQR makes the modification process faster, it may bring a larger effect on the other performance measure or cause other jobs to be tardy, so that fine-tuning rule CFR is suggested to be selected when cost is the process planning optimization target.

Cost-based Quick-tuning Rule (CQR)

Begin

(a) ……

(b) ……

(c) ……

(d) ……

(e) Solution space modification

Remove M_u from the machine set $\{M_1, M_2, \ldots, M_m\}$.

For every job in $\{J_{tdy-1}, J_{tdy-2}, \ldots, J_{tdy-n}\}$, repeat (b)-(e) until all jobs are processed

(f) ……

End

When time is the process planning optimization target, faster machine is preferred in generating an optimal process plan. This may cause the slower machine to be idle and under utilized. In this scenario, TFR is selected as the modification method and the machine is selected with relatively low utilization rate for the target job. In each round of modification, only one operation of one tardy job's solution space is modified. But for the modified operation method, the most suitable machine is identified for it and only this machine will be left as the available machine for the corresponding operation methods of the process plan solution space. The TFR is described below:

Time-based Fine-tuning Rule (TFR)

Begin

(a) ……

(b) ……

(c) ……

(d) ……

(e) Solution space modification

Check the utilization rate of each machine in {M_1, M_2, …, M_m}. Find the machine with the lowest utilization rate and assign it as M, and remove all the other machines in the machine set.

(f) ……

End

When time is the process planning optimization target, a fast tuning rule TQR is also provided. In each round of the solution space modification and in each tardy job's solution space, one operation method will be modified using the same modification method of TFR. Similar with cost-based rules, fine-tuning rule TFR is generally suggested to be selected than TQR to prevent the possible big effect to other performance measures. The TQR is described below:

Time-based Quick-tuning Rule (TQR)

Begin

(a) ……

(b) ……

(c) ……

(d) ……

(e) Solution space modification

Check the utilization rate of each machine in {M_1, M_2, …, M_m}. Find the machine with the lowest utilization rate and assign it as M, and remove all the other machines in the machine set.

For every job in {J_{tdy-1}, J_{tdy-2}, …, J_{tdy-n}}, repeat (b)-(e) until all jobs are processed

(f) ……

End

5.3.3 Heuristics for machine utilization balancing

To reduce the utilization rate of a particular machine, a common practice employed by a schedule planner is to select a job that uses the machine and replace this machine with other alternative machines. Since the process-planning module generally tries to assign the lower-cost machine in order to reduce overall cost, this adjustment will generally reduce the utilization of the overloaded machine. One obvious negative effect, however, is that the production cost of the selected job will increase. Therefore, a good trade-off is desired in job selection. The general modification heuristic rule for balancing machine utilization rate is briefly summarized below.

Begin

(a) Find the machine with the highest machine utilization rate and set it as M.

(b) Identify all the jobs, from {J_1, J_2, …, J_n}, that employ M in their process plans, and place them in a set {J_{M1}, J_{M2}, …, J_{Mk}}.

(c) Identify all the jobs, from {J_{M1}, J_{M2}, …, J_{Mk}}, that one or several OpMs having M as its machine have alternative OpMs that use other machines than M, and place them in a set {J_{or-1}, J_{or-2}, …, J_{or-m}}.

(d) If {J_{or-1}, J_{or-2}, …, J_{or-m}} = Null, the utilization of M cannot be reduced, J_M = Null, go to (f).

Else-if there is only one job in {J_{or-1}, J_{or-2}, …, J_{or-m}}, assign this job to J_M, go to (d).

Else *Calculate the total operation time ($T_{M\text{-}or}$) in which M can be replaced for each job in {$J_{or\text{-}1}$, $J_{or\text{-}2}$, …, $J_{or\text{-}m}$}. The job with the largest $T_{M\text{-}or}$ is assigned to J_M.*

End-if

(e) *Take the process plan solution space of J_M, delete the OpMs that have M as its machine, as long as there exist other alternative OpMs for the same OpT. The modified process plan solution space of J_M is thus obtained.*

(f) *Output J_M and its modified solution space.*

End

More explanations and discussions of machine utilization rate heuristics are described in detail in (Saravanan, 2001).

5.4 Process Plan Regeneration

After applying the selected heuristic, one or more process plan solution space is modified. The optimization algorithm of the process planning module is then re-run to generate an optimal process plan for the modified jobs. The newly generated process plans, together with those unmodified process plans, form a new process plan solution set. The information flow of this process is shown in Figure 5.3.

5.5 Rescheduling

After the process plan solution set is updated, rescheduling is done taking the new process plan solution set as the input. After the schedule is regenerated the facilitator takes over the control again and generates schedule performance measures. The improvement in the schedule performance is evaluated against the previously

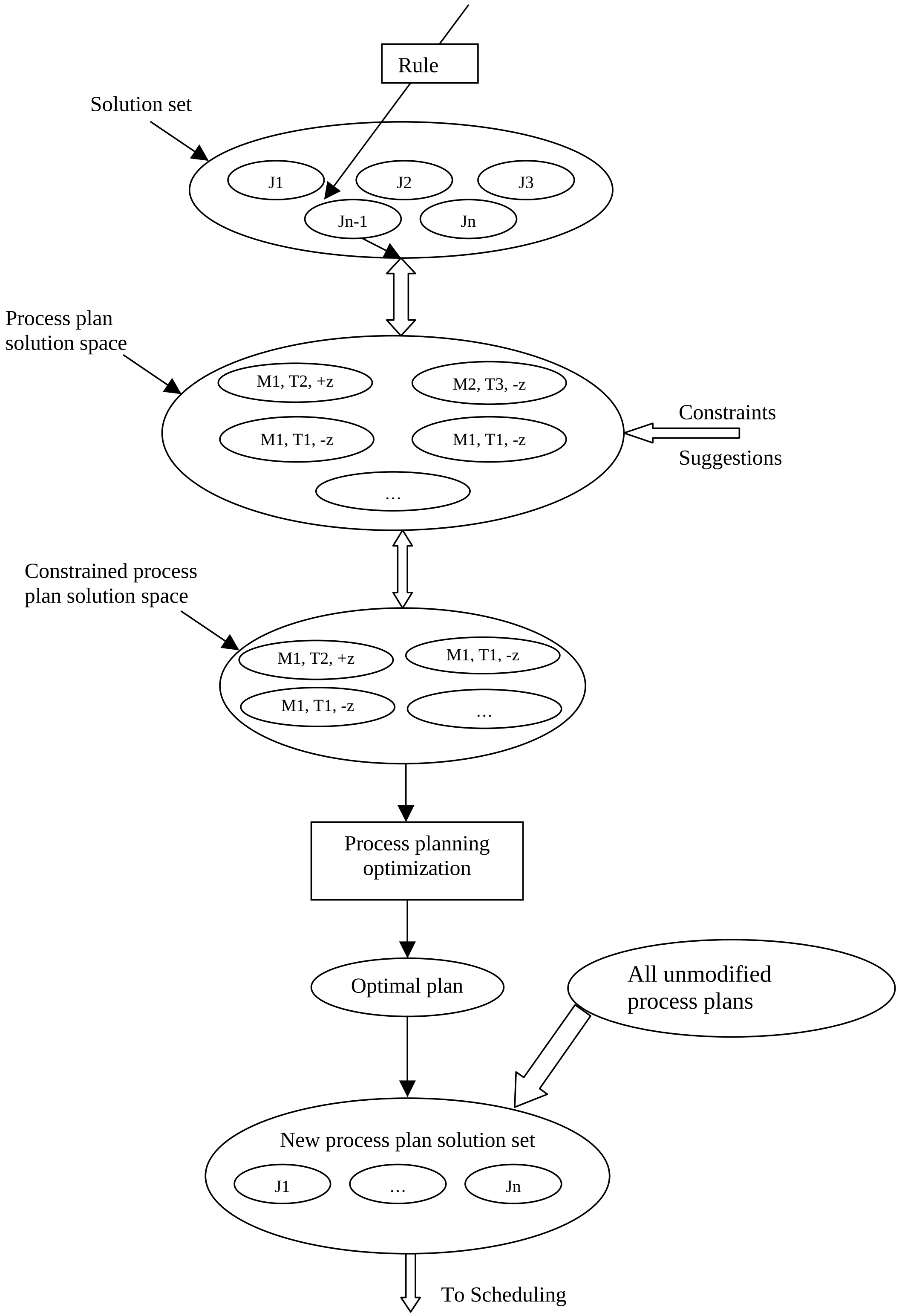


Figure 5.3 Process plan identification and modification - information flow

generated schedule's performance level. The changes are accepted if they show improvement. The improvement may be a reduction in the number of tardy jobs or the tardiness of one or more jobs. If the schedule performance is satisfactory to the user, the schedule is dispatched for production. Otherwise, the user identifies an improvement option and the whole modification process starts again.

5.6 Discussions

Both the tardy-job and machine-utilization algorithms are based on heuristics. Therefore, in terms of improving the performance measures, these algorithms are not deterministic in nature. However, since optimization is extremely hard in this case, good heuristics may serve the purpose very well. On the other hand, the user has full control on what aspects of the schedule he/she wants to improve and the improvement is achieved in a progressive manner. This could effectively prevent the scenario in which the improvement on one performance measure results the worsening of other performance measures. Actually, the author observed, during testing, that a change for the improvement on one performance measure (e.g., machine utilization) sometimes also resulted the improvement of the other one (e.g., number of tardy jobs).

Chapter 6

SYSTEM IMPLEMENTATION

6.1 Implementation Framework

The proposed integration system consists of several components: information input module, process plan module, scheduling module, facilitator module and database. The overall structure is shown in Figure 6.1. The database stores and retrieves all the information of machines, tools, and jobs. Each of the three modules, CAPP module, scheduling module and the facilitator, obtains necessary information from the database and sends its output to the database. The solution space, which stores all the available operation methods of the jobs, receives modification suggestions from facilitator. So that the database and the solution space together realize the data communication between the three modules. The main user interface is shown in Figure 6.2, which realizes the communication between the user and the integration system. The system information and the result from each module can be viewed through the user interface. The detailed description for the menu contents of the three modules will be given in the following sections.

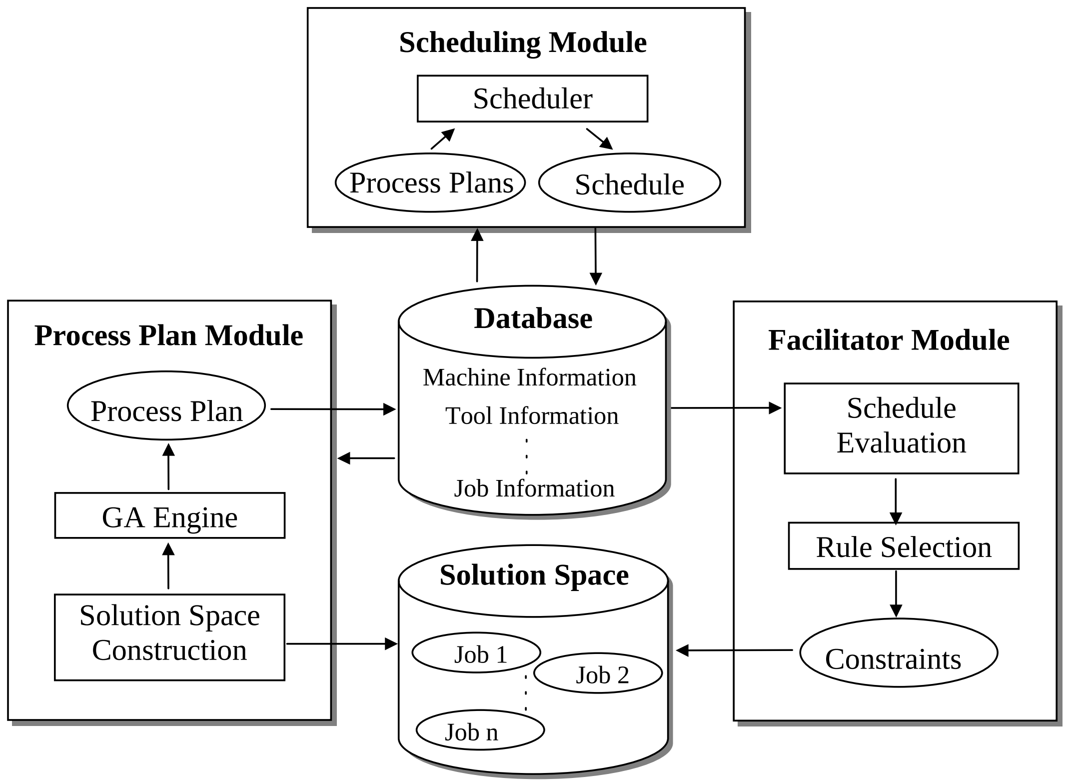


Figure 6.1 Implementation framework

6.2 CAPP Module

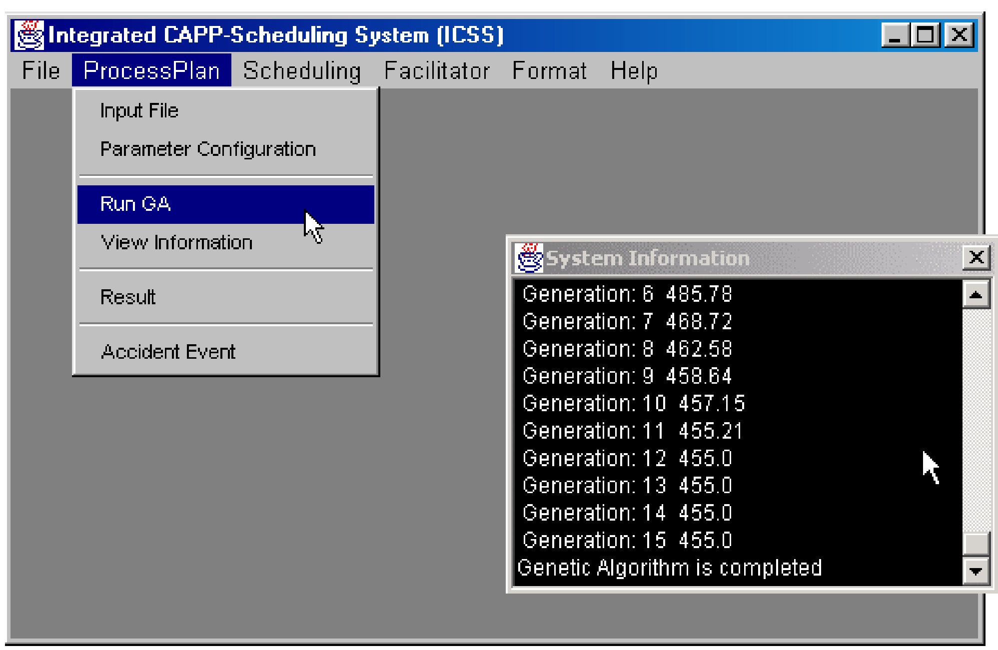


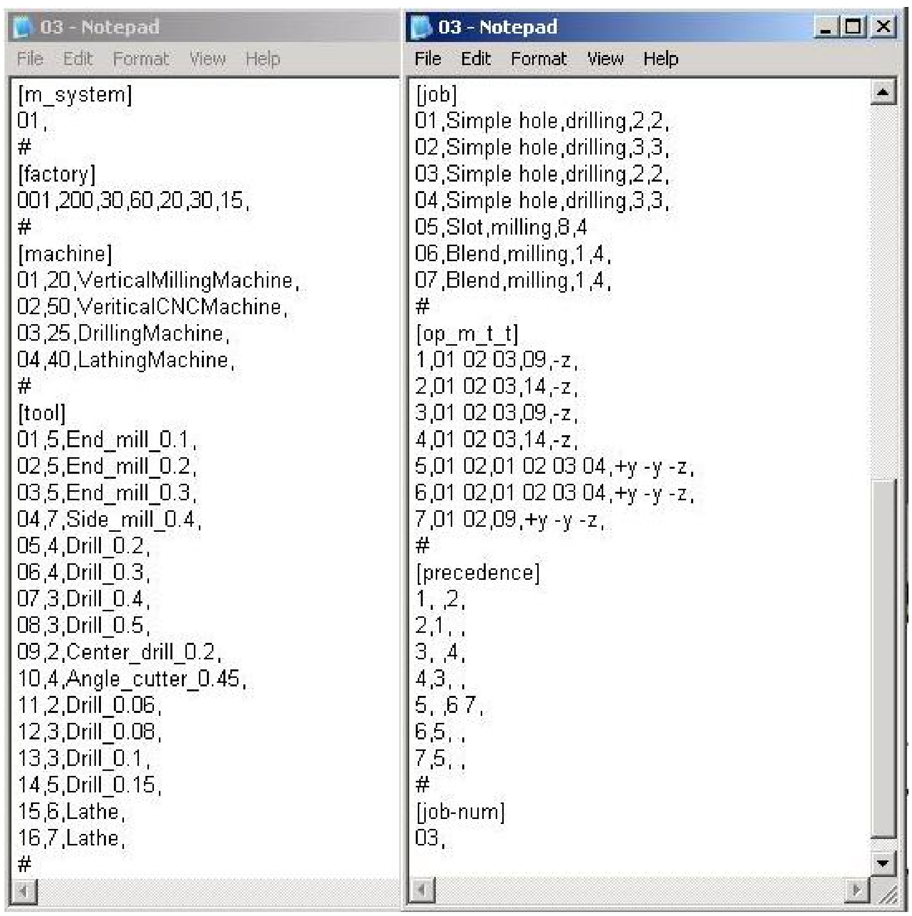
Figure 6.2 The main user interface

In the process planning module, the manufacturing information is stored in the database. The jobs to be planned and scheduled are input one at a time. The manufacturing information required for a job is described as follows:

(1) Factory information comprises the factory ID, machine information and tool information (see Figure 6.3a).

(2) Part information includes the *type* and *id* of the features and operations, OpT and OpM, etc. (see Figure 6.3b).

(3) Precedence relations between OpMs, which is also shown in Figure 6.3b.



(a) (b)

Figure 6.3 An example of process plan input file

Figure 6.2 shows the process planning module interface and the window for viewing optimization process information. After the GA optimization process, the optimal process plan is generated. A process plan result file is shown in Figure 6.4.

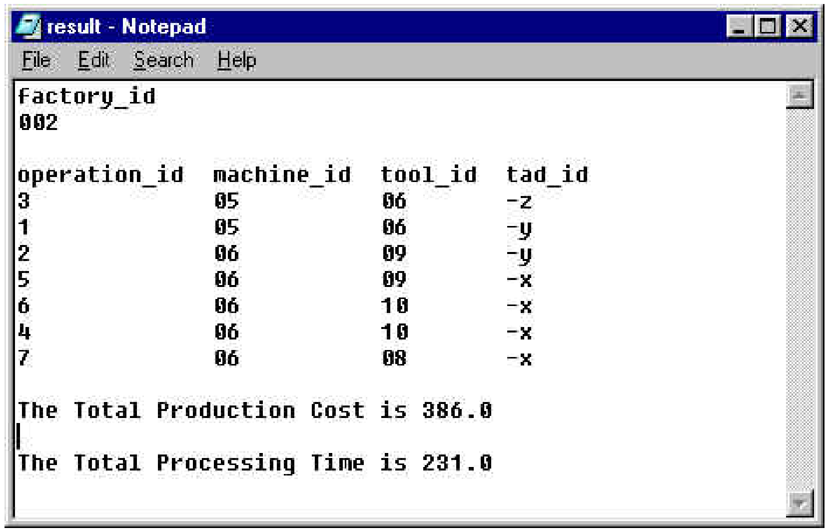


Figure 6.4 An example of process plan result file

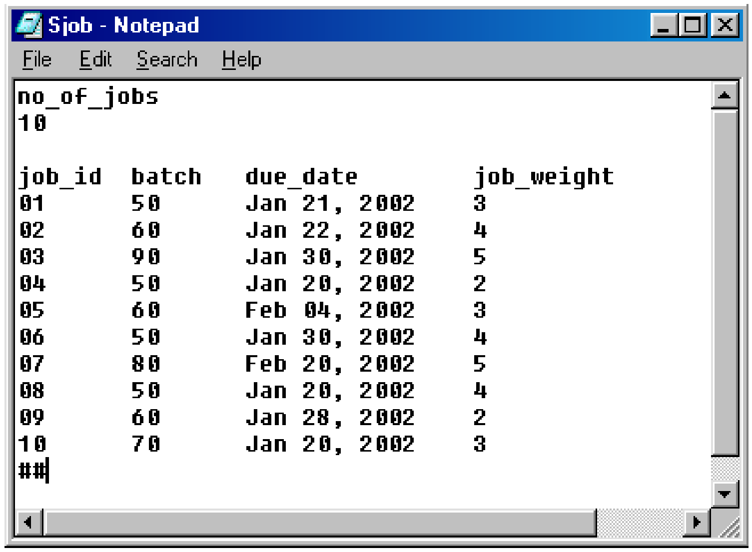
6.3 Scheduling Module

Before running the scheduling module, the following information needs to be input or selected:

(1) Process plans of the all jobs to be scheduled. The scheduling module obtains the plan information from the database.

(2) Job task information includes the due-date, job weightage, batch size, etc. The interface for data input is displayed in Figure 6.5.

(3) Scheduling strategy needs to be selected before the scheduling runs, which is shown in Figure 6.6. One of the three strategies can be selected: EDD (Earliest due date), SPT (Shorted processing time), and Weight (Job weights).



Sjob - Notepad

File Edit Search Help

```
no_of_jobs
10

job_id  batch   due_date        job_weight
01      50      Jan 21, 2002    3
02      60      Jan 22, 2002    4
03      90      Jan 30, 2002    5
04      50      Jan 20, 2002    2
05      60      Feb 04, 2002    3
06      50      Jan 30, 2002    4
07      80      Feb 20, 2002    5
08      50      Jan 20, 2002    4
09      60      Jan 28, 2002    2
10      70      Jan 20, 2002    3
##
```

Figure 6.5 An example of job information input file

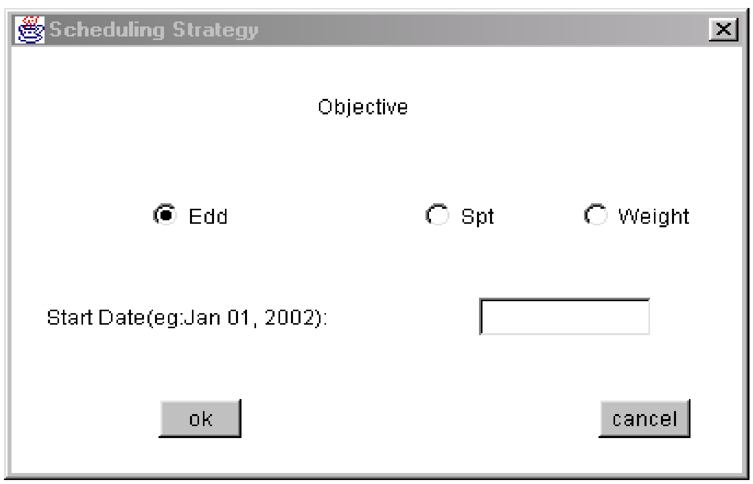


Figure 6.6 Scheduling strategy selection interface

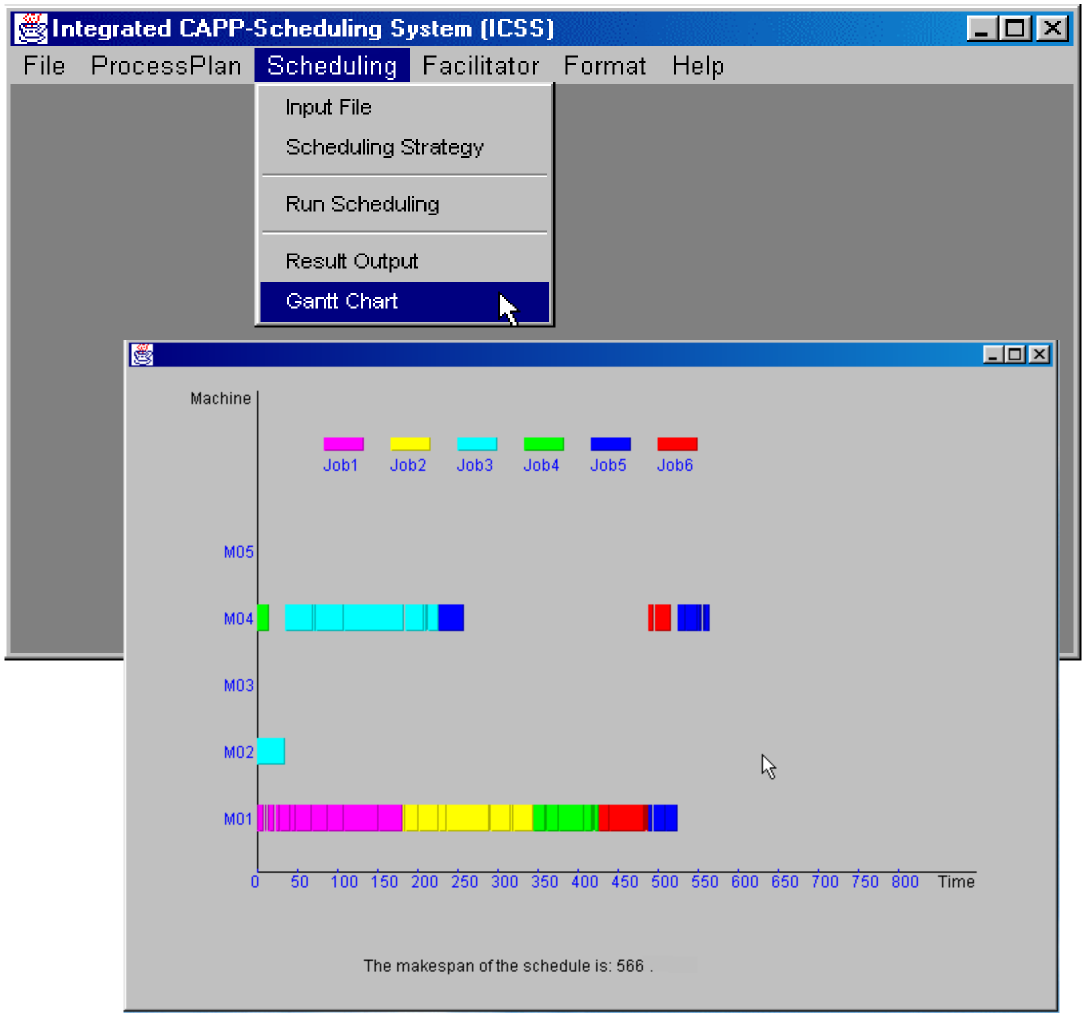


Figure 6.7 Scheduling interface and Gantt chart

The scheduling result can be output as a text file and/or Gantt chart, Figure 6.7 displays the scheduling interface and a scheduling result in the format of a Gantt chart.

6.4 Facilitator Module

The facilitator module begins with an evaluation of the schedule generated by computing its performance measures such as machine utilization and job tardiness. The performance of all the resources and jobs in the shop floor are presented graphically, as shown in Figure 6.8. The user interface allows the user to select any one of the performance measures for improvement.

After the objective is selected, the system provides two choices: the user may let the system select one suitable heuristic rule and carry out modification process automatically; or the user can select one particular rule by himself/herself.

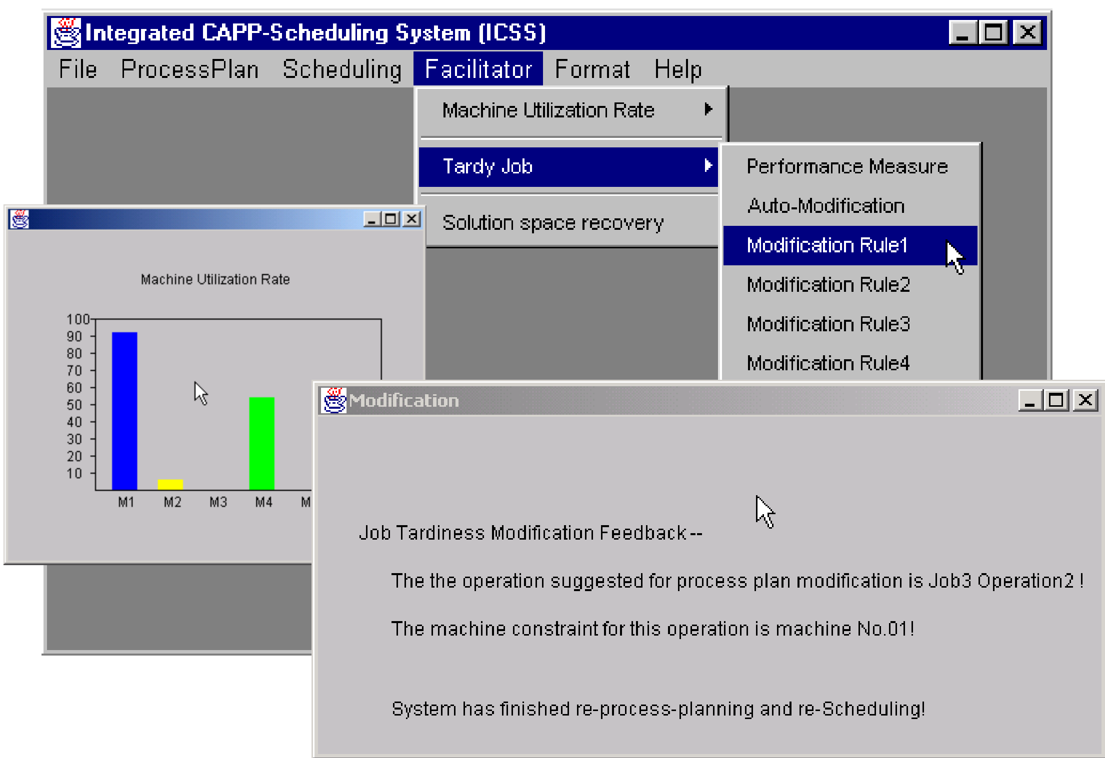


Figure 6.8 Facilitator interface

In case the modification result is not satisfactory or a user wants to retain the original process plans and schedule, the user can select the solution space recovery option so that the process planning solution space will be the same as it is before the modification process. This choice makes the facilitator algorithm more flexible.

Chapter 7

CASE STUDY

To validate the capability of the developed ICSS and illustrate how the system works, two case studies are presented in this chapter. In section 7.1, an example for minimizing job tardiness is given. In section 7.2, the comparison of using different heuristic rules is shown.

7.1 Case Study 1

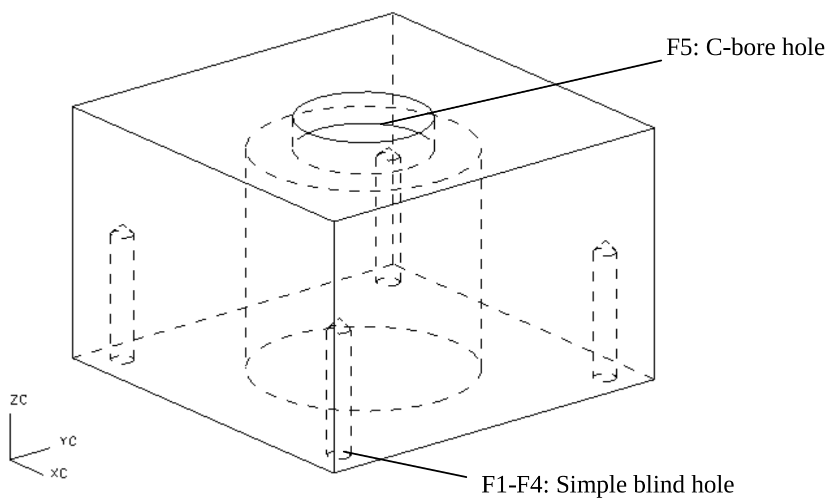
In this case, a set of eight parts to be processed in a job shop is considered and the heuristic algorithm of reducing the job tardiness is illustrated.

7.1.1 Job shop information

A job shop containing 4 machines and 16 tools is considered. The machine information and tool parameters are listed in Tables 4.1 and 4.2.

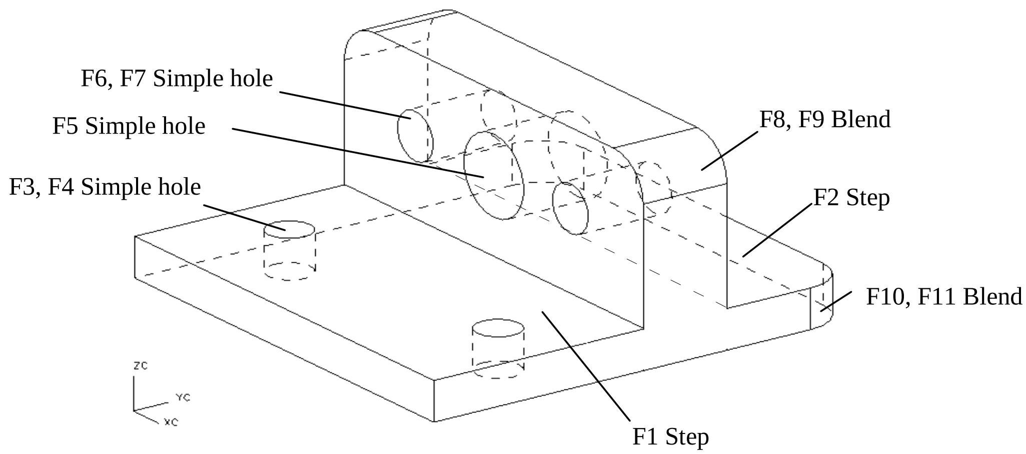
7.1.2 Example parts and the corresponding solution space

There are eight parts to be processed, and their CAD models are available. Each part is referred to as a job. After the job information is input to the process planning module, it generates the solution space for each job by assuming all the machines and tools are available and finds the optimal process plan for manufacturing the part. The parts and the corresponding process plan solution space containing the possible combinations of machines, tools, and tool access direction are presented in Figure 7.1 to Figure 7.8.



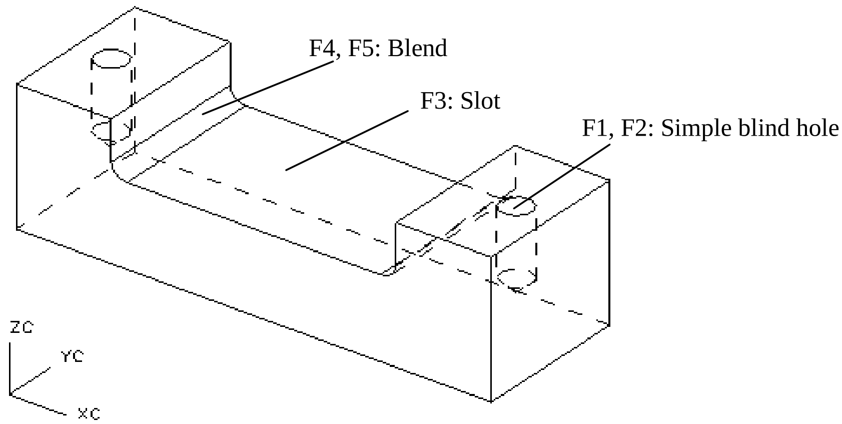
OpT	Feature	OpMs (M, T, TAD)
OpT1	F1: Simple blind hole	(M1,T9,+z) (M2,T9,+z) (M3,T9,+z)
OpT2		(M1,T11,+z) (M2,T11,+z) (M3,T11,+z)
OpT3	F2: Simple blind hole	(M1,T9,+z) (M2,T9,+z) (M3,T9,+z)
OpT4		(M1,T11,+z) (M2,T11,+z) (M3,T11,+z)
OpT5	F3: Simple blind hole	(M1,T9,+z) (M2,T9,+z) (M3,T9,+z)
OpT6		(M1,T11,+z) (M2,T11,+z) (M3,T11,+z)
OpT7	F4: Simple blind hole	(M1,T9,+z) (M2,T9,+z) (M3,T9,+z)
OpT8		(M1,T11,+z) (M2,T11,+z) (M3,T11,+z)
OpT9	F5: C_bore hole	(M1,T9,+z) (M2,T9,+z) (M3,T9,+z) (M1,T9,-z) (M2,T9,-z) (M3,T9,-z)
OpT10		(M1,T6,+z) (M2,T6,+z) (M3,T6,+z) (M1,T6,-z) (M2,T6,-z) (M3,T6,-z)
OpT11		(M1,T8,+z) (M2,T8,+z) (M3,T8,+z)

Figure 7.1 Part 1 and its process plan solution space



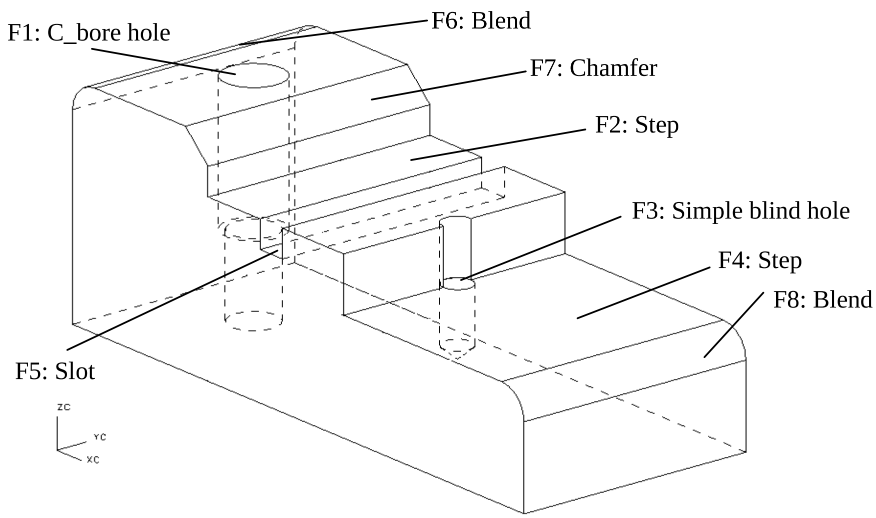
OpT	Feature	OpMs (M, T, TAD)
OpT1	F1: Step	(M1,T1,+x) (M1,T1,-x) (M1,T1,+y) (M1,T1,-z) (M1,T2,+x) (M1,T2,-x) (M1,T2,+y) (M1,T2,-z) (M1,T3,+x) (M1,T3,-x) (M1,T3,+y) (M1,T3,-z) (M1,T4,+x) (M1,T4,-x) (M1,T4,+y) (M1,T4,-z) (M2,T1,+x) (M2,T1,-x) (M2,T1,+y) (M2,T1,-z) (M2,T2,+x) (M2,T2,-x) (M2,T2,+y) (M2,T2,-z) (M2,T3,+x) (M2,T3,-x) (M2,T3,+y) (M2,T3,-z) (M2,T4,+x) (M2,T4,-x) (M2,T4,+y) (M2,T4,-z)
OpT2	F2: Step	(M1,T1,+x) (M1,T1,-x) (M1,T1,-y) (M1,T1,-z) (M1,T2,+x) (M1,T2,-x) (M1,T2,-y) (M1,T2,-z) (M1,T3,+x) (M1,T3,-x) (M1,T3,-y) (M1,T3,-z) (M1,T4,+x) (M1,T4,-x) (M1,T4,-y) (M1,T4,-z) (M2,T1,+x) (M2,T1,-x) (M2,T1,-y) (M2,T1,-z) (M2,T2,+x) (M2,T2,-x) (M2,T2,-y) (M2,T2,-z) (M2,T3,+x) (M2,T3,-x) (M2,T3,-y) (M2,T3,-z) (M2,T4,+x) (M2,T4,-x) (M2,T4,-y) (M2,T4,-z)
OpT3	F3: Simple hole	(M1,T9,-z) (M1,T9,+z) (M2,T9,-z) (M2,T9,+z)(M3,T9,-z) (M3,T9,+z)
OpT4		(M1,T14,-z) (M1,T14,+z) (M2,T14,-z) (M2,T14,+z)(M3,T14,-z) (M3,T14,+z)
OpT5	F4: Simple hole	(M1,T9,-z) (M1,T9,+z) (M2,T9,-z) (M2,T9,+z)(M3,T9,-z) (M3,T9,+z)
OpT6		(M1,T14,-z) (M1,T14,+z) (M2,T14,-z) (M2,T14,+z)(M3,T14,-z) (M3,T14,+z)
OpT7	F5:Simple hole	(M1,T9,-y) (M1,T9,+y) (M2,T9,-y) (M2,T9,+y)(M3,T9,-y) (M3,T9,+y)
OpT8		(M1,T6,-y) (M1,T6,+y) (M2,T6,-y) (M2,T6,+y)(M3,T6,-y) (M3,T6,+y)
OpT9	F6: Simple hole	(M1,T9,-y) (M1,T9,+y) (M2,T9,-y) (M2,T9,+y)(M3,T9,-y) (M3,T9,+y)
OpT10		(M1,T14,-y) (M1,T14,+y) (M2,T14,-y) (M2,T14,+y)(M3,T14,-y) (M3,T14,+y)
OpT11	F7: Simple hole	(M1,T9,-y) (M1,T9,+y) (M2,T9,-y) (M2,T9,+y)(M3,T9,-y) (M3,T9,+y)
OpT12		(M1,T14,-y) (M1,T14,+y) (M2,T14,-y) (M2,T14,+y)(M3,T14,-y) (M3,T14,+y)
OpT13	F8: Blend	(M1,T1,-y) (M1,T1,+y) (M1,T2,-y) (M1,T2,+y) (M1,T3,-y) (M1,T3,+y) (M1,T4,-y) (M1,T4,+y) (M2,T1,-y) (M2,T1,+y) (M2,T2,-y) (M2,T2,+y) (M2,T3,-y) (M2,T3,+y) (M2,T4,-x) (M2,T4,+y) (M1,T1,-z) (M1,T2,-z) (M1,T3,-z) (M1,T4,-z) (M2,T1,-z) (M2,T2,-z) (M2,T3,-z) (M2,T4,-z)
OpT14	F9: Blend	(M1,T1,-y) (M1,T1,+y) (M1,T2,-y) (M1,T2,+y) (M1,T3,-y) (M1,T3,+y) (M1,T4,-y) (M1,T4,+y) (M2,T1,-y) (M2,T1,+y) (M2,T2,-y) (M2,T2,+y) (M2,T3,-y) (M2,T3,+y) (M2,T4,-x) (M2,T4,+y) (M1,T1,-z) (M1,T2,-z) (M1,T3,-z) (M1,T4,-z) (M2,T1,-z) (M2,T2,-z) (M2,T3,-z) (M2,T4,-z)
OpT15	F10: Blend	(M1,T1,-z) (M1,T1,+z) (M1,T2,-z) (M1,T2,+z) (M1,T3,-z) (M1,T3,+z) (M1,T4,-z) (M1,T4,+z) (M1,T5,-z) (M1,T5,+z) (M2,T1,-z) (M2,T1,+z) (M2,T2,-z) (M2,T2,+z) (M2,T3,-z) (M2,T3,+z) (M2,T4,-z) (M2,T4,+z) (M2,T5,-z) (M2,T5,+z)
OpT16	F11: Blend	(M1,T1,-z) (M1,T1,+z) (M1,T2,-z) (M1,T2,+z) (M1,T3,-z) (M1,T3,+z) (M1,T4,-z) (M1,T4,+z) (M1,T5,-z) (M1,T5,+z) (M2,T1,-z) (M2,T1,+z) (M2,T2,-z) (M2,T2,+z) (M2,T3,-z) (M2,T3,+z) (M2,T4,-z) (M2,T4,+z) (M2,T5,-z) (M2,T5,+z)

Figure 7.2 Part 2 and its process plan solution space



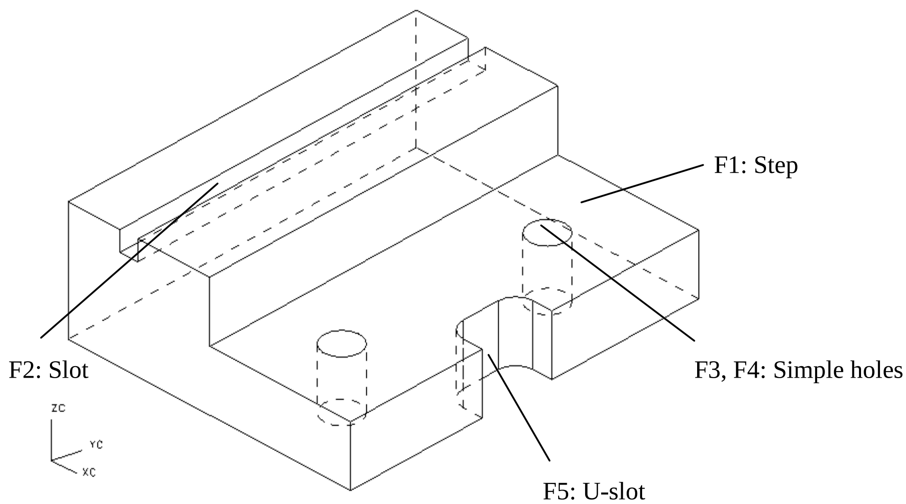
OpT	Feature	OpMs (M, T, TAD)
OpT1	F1: Simple blind hole	(M1,T9,-z) (M2,T9,-z) (M3,T9,-z)
OpT2		(M1,T14,-z) (M2,T14,-z) (M3,T14,-z)
OpT3	F2: Simple blind hole	(M1,T9,-z) (M2,T9,-z) (M3,T9,-z)
OpT4		(M1,T14,-z) (M2,T14,-z) (M3,T14,-z)
OpT5	F3: Slot	(M1,T1,-y) (M1,T1,+y) (M1,T1,-z) (M1,T2,-y) (M1,T2,+y) (M1,T2,-z) (M1,T3,-y) (M1,T3,+y) (M1,T3,-z) (M1,T4,-y) (M1,T4,+y) (M1,T4,-z) (M2,T1,-y) (M2,T1,+y) (M2,T1,-z) (M2,T2,-y) (M2,T2,+y) (M2,T2,-z) (M2,T3,-y) (M2,T3,+y) (M2,T3,-z) (M2,T4,-y) (M2,T4,+y) (M2,T4,-z)
OpT6	F4: Blend	(M1,T1,-y) (M1,T1,+y) (M1,T1,-z) (M1,T2,-y) (M1,T2,+y) (M1,T2,-z) (M1,T3,-y) (M1,T3,+y) (M1,T3,-z) (M1,T4,-y) (M1,T4,+y) (M1,T4,-z) (M2,T1,-y) (M2,T1,+y) (M2,T1,-z) (M2,T2,-y) (M2,T2,+y) (M2,T2,-z) (M2,T3,-y) (M2,T3,+y) (M2,T3,-z) (M2,T4,-y) (M2,T4,+y) (M2,T4,-z)
OpT7	F5: Blend	(M1,T1,-y) (M1,T1,+y) (M1,T1,-z) (M1,T2,-y) (M1,T2,+y) (M1,T2,-z) (M1,T3,-y) (M1,T3,+y) (M1,T3,-z) (M1,T4,-y) (M1,T4,+y) (M1,T4,-z) (M2,T1,-y) (M2,T1,+y) (M2,T1,-z) (M2,T2,-y) (M2,T2,+y) (M2,T2,-z) (M2,T3,-y) (M2,T3,+y) (M2,T3,-z) (M2,T4,-y) (M2,T4,+y) (M2,T4,-z)

Figure 7.3 Part 3 and its process plan solution space



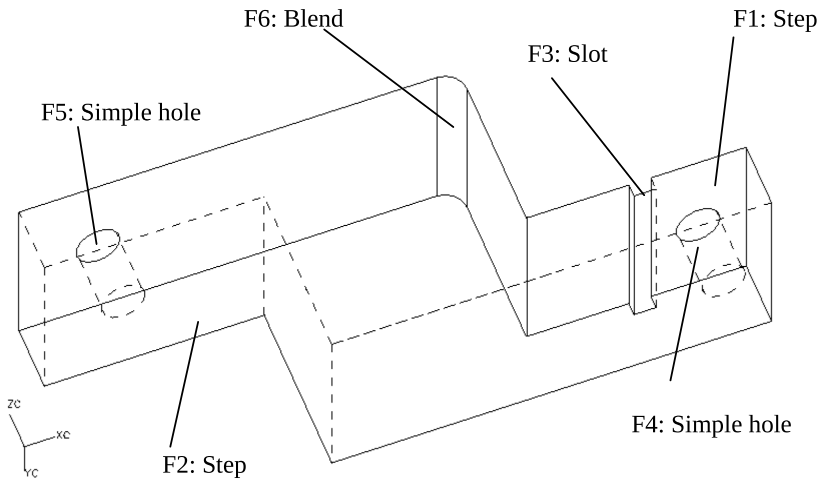
OpT	Feature	OpMs (M, T, TAD)
OpT1	F1: C_bore hole	(M1,T9,+z) (M1,T9,-z) (M2,T9,-z) (M2,T9,+z) (M3,T9,-z) (M3,T9,+z)
OpT2		(M1,T5,+z) (M2,T5,+z) (M3,T5,+z)
OpT3		(M1,T6,-z) (M2,T6,-z) (M3,T6,-z)
OpT4	F2: Step	(M1,T1,+x) (M1,T1,-x) (M1,T1,+y) (M1,T1,-z) (M1,T2,+x) (M1,T2,-x) (M1,T2,+y) (M1,T2,-z) (M1,T3,+x) (M1,T3,-x) (M1,T3,+y) (M1,T3,-z) (M1,T4,+x) (M1,T4,-x) (M1,T4,+y) (M1,T4,-z) (M2,T1,+x) (M2,T1,-x) (M2,T1,+y) (M2,T1,-z) (M2,T2,+x) (M2,T2,-x) (M2,T2,+y) (M2,T2,-z) (M2,T3,+x) (M2,T3,-x) (M2,T3,+y) (M2,T3,-z) (M2,T4,+x) (M2,T4,-x) (M2,T4,+y) (M2,T4,-z)
OpT5	F3: Simple blind hole	(M1,T9,-z) (M2,T9,-z) (M3,T9,-z)
OpT6		(M1,T12,-z) (M2,T12,-z) (M3,T12,-z)
OpT7	F4: Step	(M1,T1,+x) (M1,T1,-x) (M1,T1,+y) (M1,T1,-z) (M1,T2,+x) (M1,T2,-x) (M1,T2,+y) (M1,T2,-z) (M1,T3,+x) (M1,T3,-x) (M1,T3,+y) (M1,T3,-z) (M1,T4,+x) (M1,T4,-x) (M1,T4,+y) (M1,T4,-z) (M2,T1,+x) (M2,T1,-x) (M2,T1,+y) (M2,T1,-z) (M2,T2,+x) (M2,T2,-x) (M2,T2,+y) (M2,T2,-z) (M2,T3,+x) (M2,T3,-x) (M2,T3,+y) (M2,T3,-z) (M2,T4,+x) (M2,T4,-x) (M2,T4,+y) (M2,T4,-z)
OpT8	F5: Slot	(M1,T1,-z) (M2,T1,-z)
OpT9	F6: Blend	(M1,T1,-y) (M1,T1,+y) (M1,T1,-z) (M1,T2,-y) (M1,T2,+y) (M1,T2,-z) (M1,T3,-y) (M1,T3,+y) (M1,T3,-z) (M1,T4,-y) (M1,T4,+y) (M1,T4,-z) (M2,T1,-y) (M2,T1,+y) (M2,T1,-z) (M2,T2,-y) (M2,T2,+y) (M2,T2,-z) (M2,T3,-y) (M2,T3,+y) (M2,T3,-z) (M2,T4,-y) (M2,T4,+y) (M2,T4,-z)
OpT10	F7: Chamfer	(M1,T1,-y) (M1,T1,+y) (M1,T2,-y) (M1,T2,+y) (M1,T3,-y) (M1,T3,+y) (M1,T4,-y) (M1,T4,+y) (M2,T1,-y) (M2,T1,+y) (M2,T2,-y) (M2,T2,+y) (M2,T3,-y) (M2,T3,+y) (M2,T4,-y) (M2,T4,+y)
OpT11	F8: Blend	(M1,T1,-y) (M1,T1,+y) (M1,T1,-z) (M1,T2,-y) (M1,T2,+y) (M1,T2,-z) (M1,T3,-y) (M1,T3,+y) (M1,T3,-z) (M1,T4,-y) (M1,T4,+y) (M1,T4,-z) (M2,T1,-y) (M2,T1,+y) (M2,T1,-z) (M2,T2,-y) (M2,T2,+y) (M2,T2,-z) (M2,T3,-y) (M2,T3,+y) (M2,T3,-z) (M2,T4,-y) (M2,T4,+y) (M2,T4,-z)

Figure 7.4 Part 4 and its process plan solution space



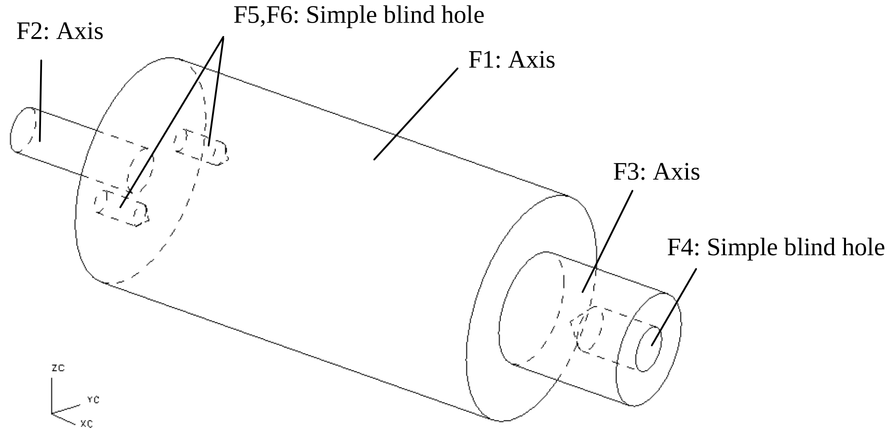
OpT	Feature	OpMs (M, T, TAD)
OpT1	F1: Step	(M1,T1,+y) (M1,T1,-y) (M1,T1,-x) (M1,T1,-z) (M1,T2,+y) (M1,T2,-y) (M1,T2,-x) (M1,T2,-z) (M1,T3,+y) (M1,T3,-y) (M1,T3,-x) (M1,T3,-z) (M1,T4,+y) (M1,T4,-y) (M1,T4,-x) (M1,T4,-z) (M2,T1,+y) (M2,T1,-y) (M2,T1,-x) (M2,T1,-z) (M2,T2,+y) (M2,T2,-y) (M2,T2,-x) (M2,T2,-z) (M2,T3,+y) (M2,T3,-y) (M2,T3,-x) (M2,T3,-z) (M2,T4,+y) (M2,T4,-y) (M2,T4,-x) (M2,T4,-z)
OpT2	F2: Slot	(M1,T1,-z) (M2,T1,-z)
OpT3	F3: Simple hole	(M1,T9,-z) (M2,T9,-z) (M3,T9,-z) (M1,T9,+z) (M2,T9,+z) (M3,T9,+z)
OpT4		(M1,T5,-z) (M2,T5,-z) (M3,T5,-z) (M1,T5,+z) (M2,T5,+z) (M3,T5,+z)
OpT5	F4: Simple hole	(M1,T9,-z) (M2,T9,-z) (M3,T9,-z) (M1,T9,+z) (M2,T9,+z) (M3,T9,+z)
OpT6		(M1,T5,-z) (M2,T5,-z) (M3,T5,-z) (M1,T5,+z) (M2,T5,+z) (M3,T5,+z)
OpT7	F5: U-slot	(M1,T1,-z) (M1,T1,+z) (M1,T1,-x) (M1,T2,-z) (M1,T2,+z) (M1,T2,-x) (M1,T3,-z) (M1,T3,+z) (M1,T3,-x) (M2,T1,-z) (M2,T1,+z) (M2,T1,-x) (M2,T2,-z) (M2,T2,+z) (M2,T2,-x) (M2,T3,-z) (M2,T3,+z) (M2,T3,-x)

Figure 7.5 Part 5 and its process plan solution space



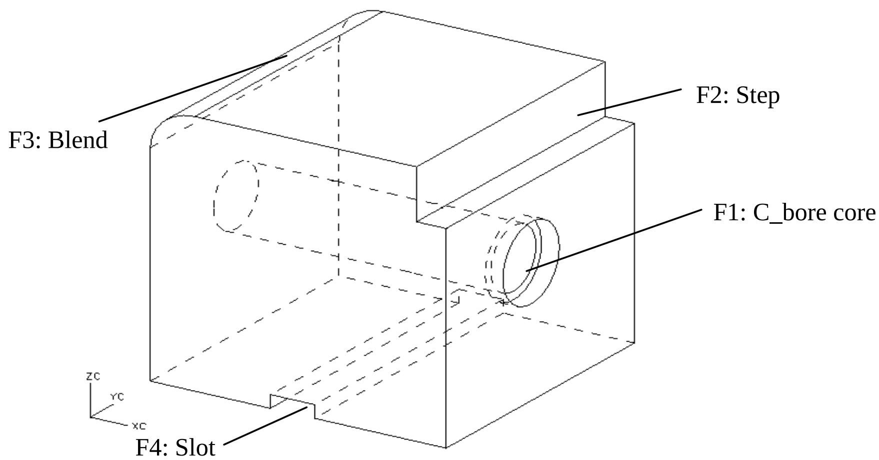
OpT	Feature	OpMs (M, T, TAD)
OpT1	F1: Step	(M1,T1,+y) (M1,T1,-y) (M1,T1,-x) (M1,T1,-z) (M1,T2,+y) (M1,T2,-y) (M1,T2,-x) (M1,T2,-z) (M1,T3,+y) (M1,T3,-y) (M1,T3,-x) (M1,T3,-z) (M1,T4,+y) (M1,T4,-y) (M1,T4,-x) (M1,T4,-z) (M2,T1,+y) (M2,T1,-y) (M2,T1,-x) (M2,T1,-z) (M2,T2,+y) (M2,T2,-y) (M2,T2,-x) (M2,T2,-z) (M2,T3,+y) (M2,T3,-y) (M2,T3,-x) (M2,T3,-z) (M2,T4,+y) (M2,T4,-y) (M2,T4,-x) (M2,T4,-z)
OpT2	F2: Step	(M1,T1,+y) (M1,T1,-y) (M1,T1,+x) (M1,T1,+z) (M1,T2,+y) (M1,T2,+y) (M1,T2,+x) (M1,T2,+z) (M1,T3,+y) (M1,T3,-y) (M1,T3,+x) (M1,T3,+z) (M1,T4,+y) (M1,T4,-y) (M1,T4,+x) (M1,T4,+z) (M2,T1,+y) (M2,T1,-y) (M2,T1,+x) (M2,T1,+z) (M2,T2,+y) (M2,T2,-y) (M2,T2,+x) (M2,T2,+z) (M2,T3,+y) (M2,T3,-y) (M2,T3,+x) (M2,T3,+z) (M2,T4,+y) (M2,T4,-y) (M2,T4,+x) (M2,T4,+z)
OpT3	F3: Slot	(M1,T1,-z) (M2,T1,-z) (M1,T2,-z) (M2,T2,-z)
OpT4	F4: Simple hole	(M1,T9,-z) (M2,T9,-z) (M3,T9,-z) (M1,T9,+z) (M2,T9,+z) (M3,T9,+z)
OpT5		(M1,T7,-z) (M2,T7,-z) (M3,T7,-z) (M1,T7,+z) (M2,T7,+z) (M3,T7,+z)
OpT6	F5: Simple hole	(M1,T9,-z) (M2,T9,-z) (M3,T9,-z) (M1,T9,+z) (M2,T9,+z) (M3,T9,+z)
OpT7		(M1,T7,-z) (M2,T7,-z) (M3,T7,-z) (M1,T7,+z) (M2,T7,+z) (M3,T7,+z)
OpT8	F6: Blend	(M1,T1,-y) (M1,T1,+y) (M1,T1,-z) (M1,T2,-y) (M1,T2,+y) (M1,T2,-z) (M1,T3,-y) (M1,T3,+y) (M1,T3,-z) (M1,T4,-y) (M1,T4,+y) (M1,T4,-z) (M2,T1,-y) (M2,T1,+y) (M2,T1,-z) (M2,T2,-y) (M2,T2,+y) (M2,T2,-z) (M2,T3,-y) (M2,T3,+y) (M2,T3,-z) (M2,T4,-y) (M2,T4,+y) (M2,T4,-z)

Figure 7.6 Part 6 and its process plan solution space



OpT	Feature	OpMs (M, T, TAD)
OpT1	F1: Axis	(M4,T15,+x) (M4,T15,-x) (M4,T16,+x) (M4,T16,-x)
OpT2	F2: Axis	(M4,T15,+x) (M4,T16,+x)
OpT3	F3: Axis	(M4,T15,-x) (M4,T16,-x)
OpT4	F4: Simple blind hole	(M1,T9,-x) (M2,T9,-x) (M3,T9,-x)
OpT5		(M1,T6,+x) (M2,T6,+x) (M3,T6,+x)
OpT6	F5: Simple blind hole	(M1,T9,-x) (M2,T9,-x) (M3,T9,-x)
OpT7		(M1,T14,+x) (M2,T14,+x) (M3,T14,+x)
OpT8	F6: Simple blind hole	(M1,T9,-x) (M2,T9,-x) (M3,T9,-x)
OpT9		(M1,T14,+x) (M2,T14,+x) (M3,T14,+x)

Figure 7.7 Part 7 and its process plan solution space



OpT	Feature	OpMs (M, T, TAD)
OpT1	F1: C_bore hole	(M1,T9,+x) (M1,T9,-x) (M2,T9,-x) (M2,T9,+x) (M3,T9,-x) (M3,T9,+x)
OpT2		(M1,T7,+x) (M1,T7,-x) (M2,T7,+x) (M2,T7,-x) (M3,T7,+x) (M3,T7,-x)
OpT3		(M1,T8,-x) (M2,T8,-x) (M3,T8,-x)
OpT4	F2: Step	(M1,T1,+y) (M1,T1,-y) (M1,T1,-x) (M1,T1,-z) (M1,T2,+y) (M1,T2,-y) (M1,T2,-x) (M1,T2,-z) (M1,T3,+y) (M1,T3,-y) (M1,T3,-x) (M1,T3,-z) (M1,T4,+y) (M1,T4,-y) (M1,T4,-x) (M1,T4,-z) (M2,T1,+y) (M2,T1,-y) (M2,T1,-x) (M2,T1,-z) (M2,T2,+y) (M2,T2,-y) (M2,T2,-x) (M2,T2,-z) (M2,T3,+y) (M2,T3,-y) (M2,T3,-x) (M2,T3,-z) (M2,T4,+y) (M2,T4,-y) (M2,T4,-x) (M2,T4,-z)
OpT5	F3: Blend	(M1,T1,-y) (M1,T1,+y) (M1,T1,-z) (M1,T2,-y) (M1,T2,+y) (M1,T2,-z) (M1,T3,-y) (M1,T3,+y) (M1,T3,-z) (M1,T4,-y) (M1,T4,+y) (M1,T4,-z) (M2,T1,-y) (M2,T1,+y) (M2,T1,-z) (M2,T2,-y) (M2,T2,+y) (M2,T2,-z) (M2,T3,-y) (M2,T3,+y) (M2,T3,-z) (M2,T4,-y) (M2,T4,+y) (M2,T4,-z)
OpT6	F4: Slot	(M1,T1,+z) (M2,T1,+z) (M1,T2,+z) (M2,T2,+z)

Figure 7.8 Part 8 and its process plan solution space

In the GA optimization process, necessary parameters are set or selected through the user interface. In this case, minimizing processing cost is selected as the process planning optimization target. The process plans for the eight parts are generated respectively and input to the scheduling module.

7.1.3 The generation of schedule

The job order information is listed in Table 7.1, which includes batch size, due date and job weight of each job. The manufacturing start date is Jan 01, 2002.

Table 7.1 Job information

Job No.	Batch size	Due date	Job weight
01	40	Jan 23, 2002	3
02	50	Jan 15, 2002	4
03	70	Jan 19, 2002	5
04	30	Jan 21, 2002	2
05	30	Jan 24, 2002	5
06	40	Jan 26, 2002	2
07	60	Jan 21, 2002	3
08	80	Jan 22, 2002	2

After running the scheduling module, a schedule with four tardy jobs is generated, which are Job1, Job5, Job6 and Job8. The tardy job information is shown in Figure 7.9a (the unit of tardiness is 'day').

7.1.4 Constraint generation and plan solution space modification

As the process plan optimization objective is cost, the rule CFR is employed as the modification rule. After running this tardy job heuristic rule, the output result, i.e. the modification constraint was: Job8/OpM5/M1, which means the modification target OpM5 of Job8 was found, since Job8 has the lowest tardiness and OpM1 has the longest operation waiting time. Table 7.2 shows the possible M/T/TADs of OpT5 of Job8. The operation methods of OpM5 with M1 should be deleted from the solution space as shown.

Table 7.2 Solution space of Job8

OpT	Feature	OpMs (M, T, TAD)
: :	: :	: :
OpT5	F3: Blend	~~***(M1,T1,-y) (M1,T1,+y) (M1,T1,-z) (M1,T2,-y) (M1,T2,+y) (M1,T2,-z)***~~ ~~***(M1,T3,-y) (M1,T3,+y) (M1,T3,-z) (M1,T4,-y) (M1,T4,+y) (M1,T4,-z)***~~ (M2,T1,-y) (M2,T1,+y) (M2,T1,-z) (M2,T2,-y) (M2,T2,+y) (M2,T2,-z) (M2,T3,-y) (M2,T3,+y) (M2,T3,-z) (M2,T4,-y) (M2,T4,+y) (M2,T4,-z)
: :	: :	: :

After modification of the solution space, process planning and scheduling were re-run automatically. The tardy job performance of the newly generated schedule is listed in Figure 7.9b. One can see that the job tardiness of Job4, Job5 and Job6 has been reduced. After the modification process continues for four iterations, the performance measure shows all the tardy jobs have been removed. The tardy job performance measure of all the four iterations is shown in Figure 7.9.

7.1.5 Result and discussions

After four iterations, the number of tardy jobs is reduced to zero. In this process, other performance measures have also changed corresponding to the modification. The machine utilization rate changing process of the four machines is shown in Figure 7.10.

In Figure 7.10, one can see that M1 is the highest utilized machine in the whole process. It is because M1 has the smallest machine cost index and therefore is the most preferred machine to use in the process planning optimization process. This not only causes M1 to be over-utilized but also may make jobs queuing to be processed on M1 and consequently cause some jobs to be tardy. By using the tardy job modification rule, some jobs are arranged to avoid being processed on a busy machine, so that the machine utilization rate is also balanced effectively.

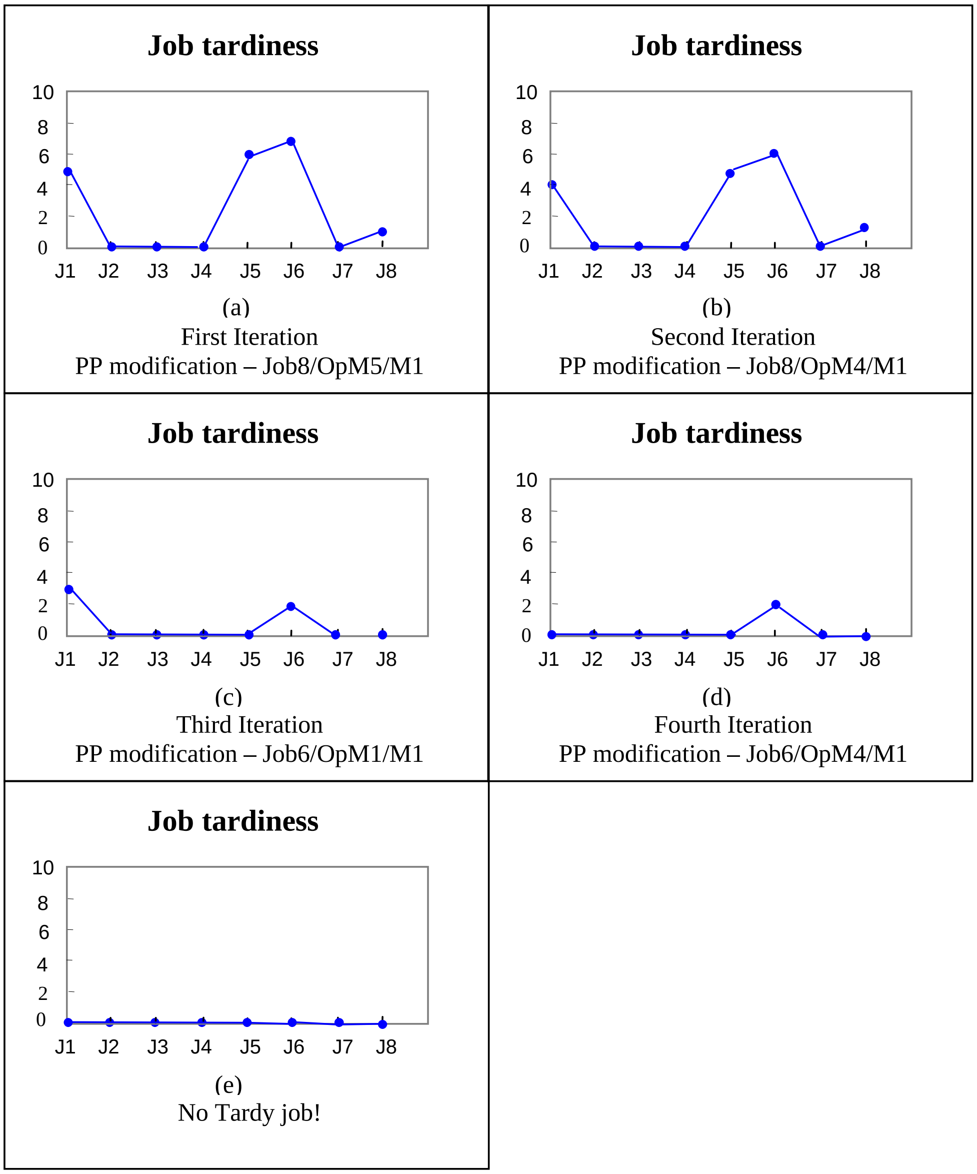


Figure 7.9 The process of reducing job tardiness

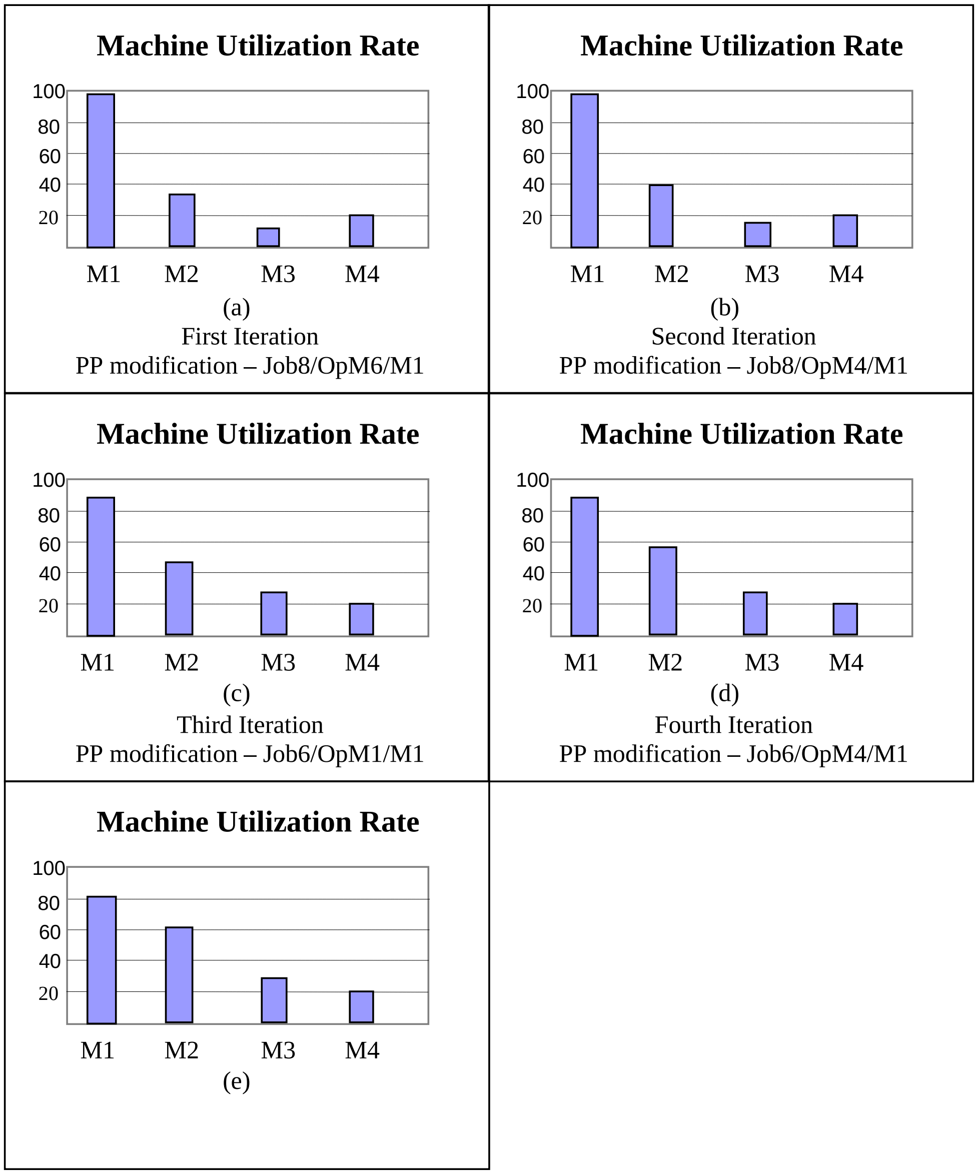


Figure 7.10 The machine utilization rate changing information

7.2 Case Study 2

For job tardiness minimization, four heuristic rules have been developed for solution space modification. The facilitator suggests the most suitable rule and applies the generated constraints to the process plan solution space. However, it does not mean that the other three rules cannot improve the selected performance measure. In this section, we use a simulated based case study, which is comprised of 15 jobs to be processed, to test and compare the results of implementing the four tardy job modification rules. Table 7.3 shows the job information. The time and cost in the table refers to the time and cost of the initially generated process plan for each job.

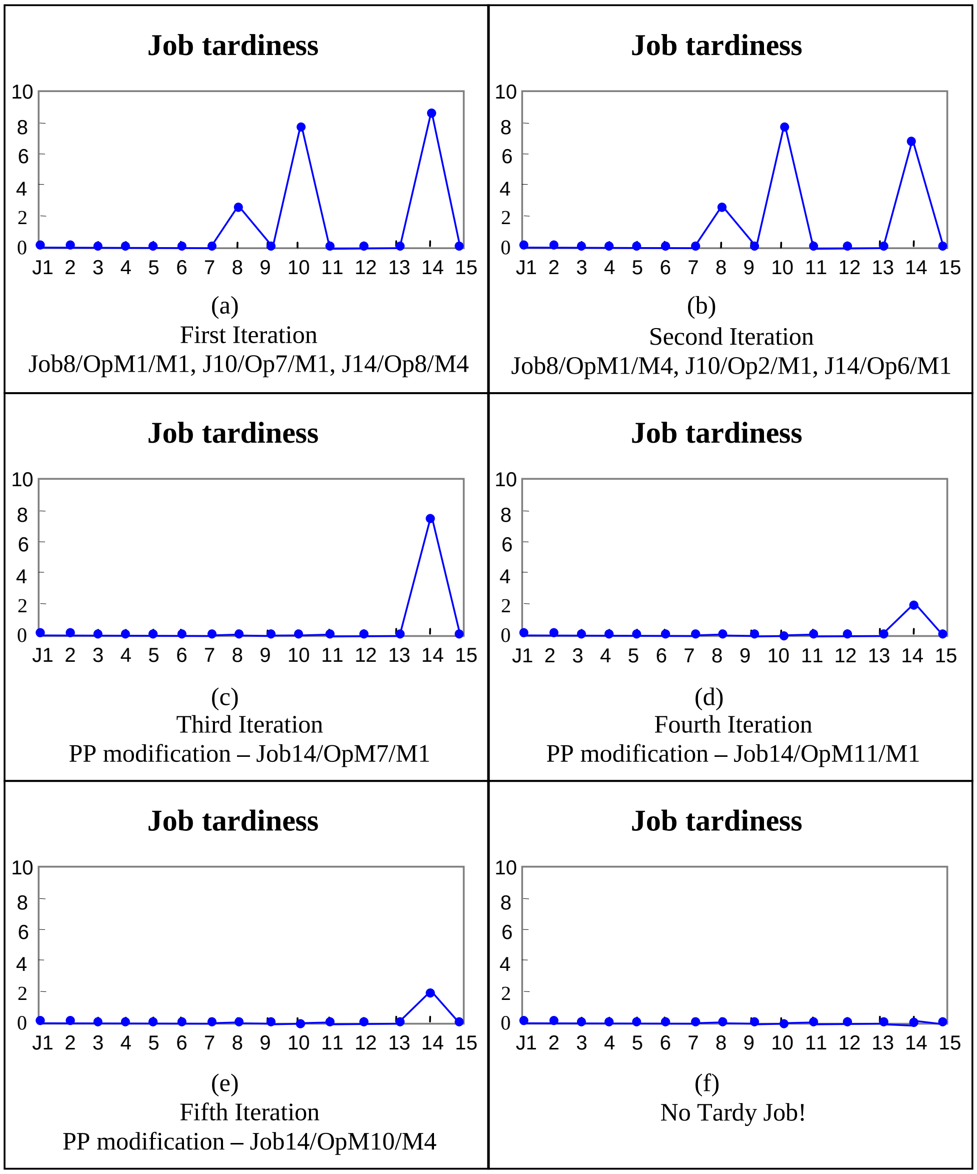
Table 7.3 Job information

Job No.	Batch size	Due date	Job weight	Time index	Cost index
01	30	Feb 05, 2002	2	273	345
02	50	Feb 07, 2002	5	353	475
03	40	Feb 04, 2002	3	259	342
04	50	Feb 10, 2002	6	525	998
05	60	Feb 09, 2002	1	331	642
06	40	Feb 06, 2002	4	333	711
07	30	Feb 09, 2002	2	327	439
08	50	Feb 24, 2002	7	483	890
09	40	Feb 26, 2002	1	339	580
10	20	Feb 27, 2002	4	374	782
11	40	Feb 19, 2002	3	302	665
12	70	Feb 15, 2002	5	302	640
13	60	Feb 03, 2002	4	374	782
14	60	Feb 25, 2002	2	393	865
15	30	Feb 21, 2002	1	461	949

After running the scheduling algorithm using the EDD heuristic, the resulted schedule has three tardy jobs: Job8, Job10 and Job14. The tardy job information is shown in Figure 7.11a. Since cost is the process planning optimization target, cost-based rules should be selected for tardy job modification. Figure 7.11 and Figure 7.12

shows the modification process using CQR (Cost-based Quick-tuning Rule) and CFR (Cost-based Fine-tuning Rule) respectively. Both of them reach a zero-tardiness schedule; CQR took five iterations whereas CFR took ten iterations.

Using time-based rules, TFR and TQR, also achieves a schedule with zero tardiness finally, which needs five iterations and eight iterations respectively. However the two time-based rules resulted in a higher cost increase compared with that of cost-based rules. Each time after the modification of the process planning solution space and re-running the optimization process, the production cost and time changes of the newly generated process plan are recorded. Figure 7.13 shows the production cost increase of the modification process using the four rules, and Figure 7.14 shows the production time information. In this case, CQR not only needs less iteration (five iterations) than CFR (ten iterations) but also results in less cost increase and less time increase. The comparison of cost increase of the four rules, which is shown in Figure 7.13, indicated that cost based rules (CFR and CQR) perform better than time based rules (TFR and THR).



(a)
First Iteration
Job8/OpM1/M1, J10/Op7/M1, J14/Op8/M4

(b)
Second Iteration
Job8/OpM1/M4, J10/Op2/M1, J14/Op6/M1

(c)
Third Iteration
PP modification – Job14/OpM7/M1

(d)
Fourth Iteration
PP modification – Job14/OpM11/M1

(e)
Fifth Iteration
PP modification – Job14/OpM10/M4

(f)
No Tardy Job!

Figure 7.11 The process of reducing job tardiness by CQR

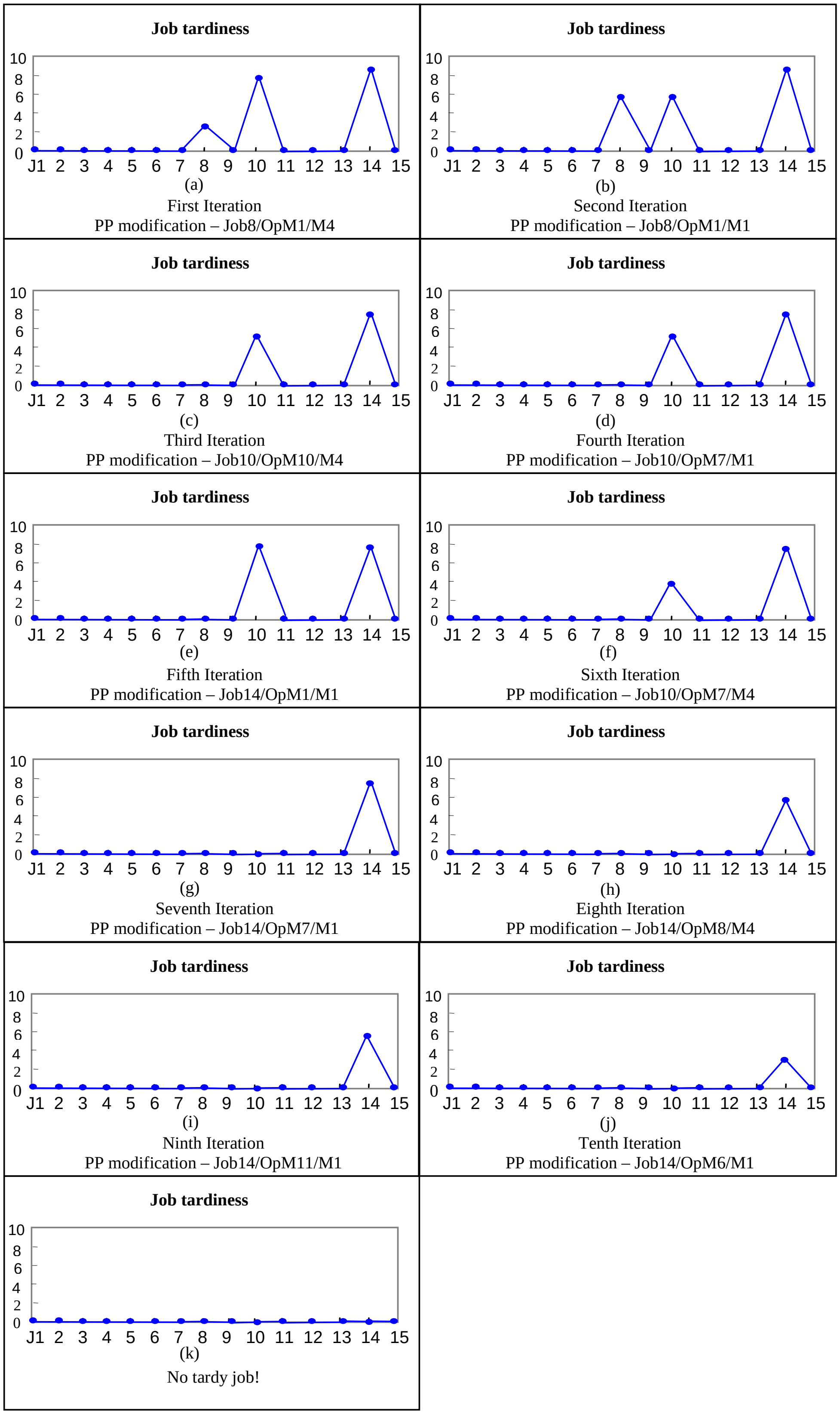


Figure 7.12 The process of reducing job tardiness by CFR

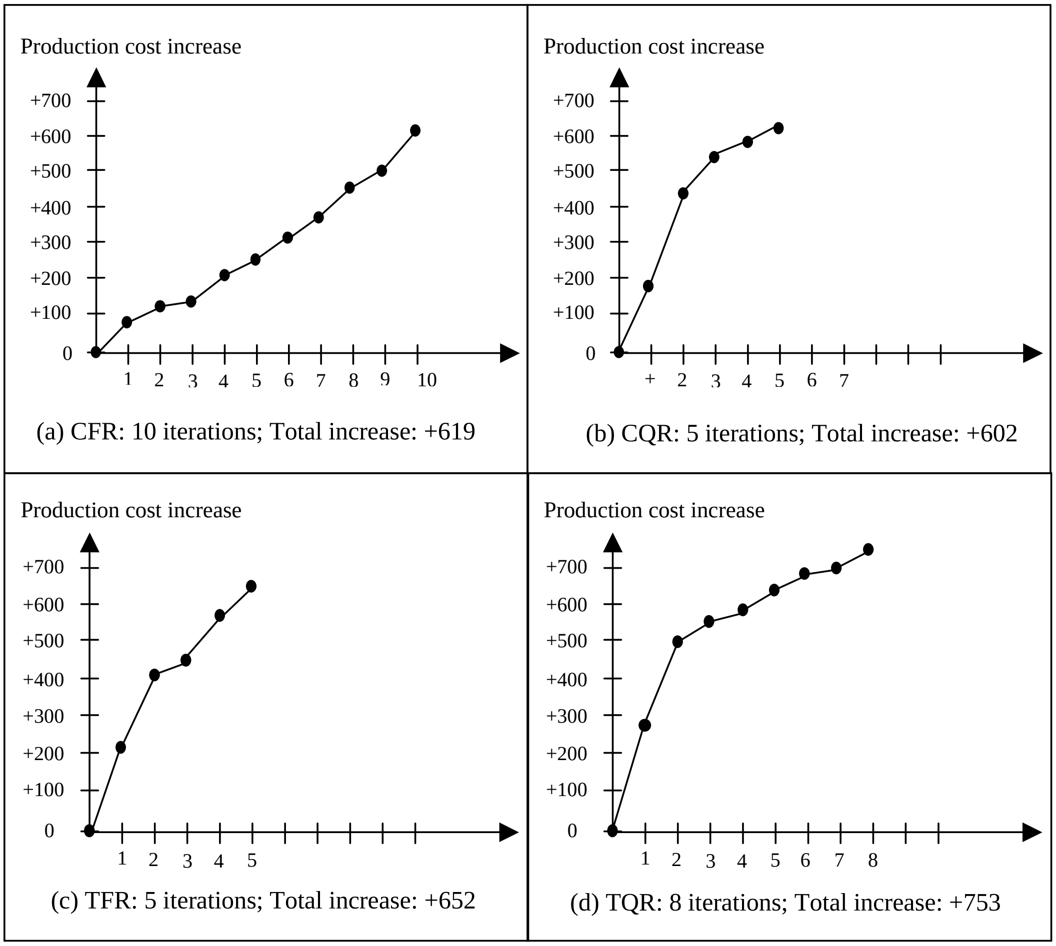
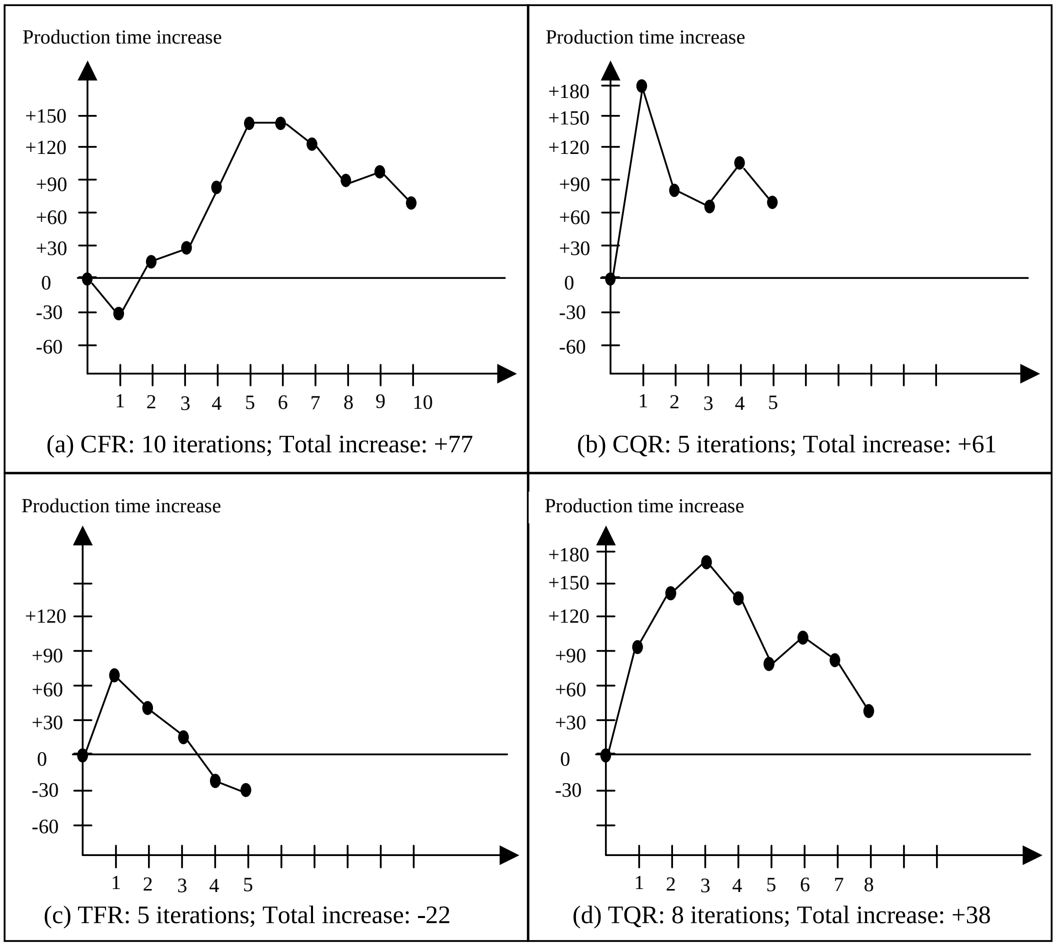


Figure 7.13 The comparison of four rules by production cost increase



(a) CFR: 10 iterations; Total increase: +77

(b) CQR: 5 iterations; Total increase: +61

(c) TFR: 5 iterations; Total increase: -22

(d) TQR: 8 iterations; Total increase: +38

Figure 7.14 The comparison of four rules by production time increase

Chapter 8

CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

A new approach towards the integration of process planning and scheduling has been proposed in this thesis, in which the flexibility of process planning is extensively explored to achieve a satisfactory schedule according to established performance measures. The system architecture and the three important modules – process planning module, scheduling module, and facilitator modules are presented. The system can handle multiple scheduling objectives and the user has the choice to select the performance measure of a schedule, which needs to be improved. Heuristic rules for balancing machine utilization rate and reducing tardy jobs have been developed. The main contributions of this research are summarized as in the following:

(1) Firstly, the facilitator module, through adding constraints to the solution space of the process planning module, realizes the integration of process planning and scheduling. As the integrator, the facilitator module not only works as the interface to realize the communication between the process planning module

and the scheduling module, but also makes the three modules cooperate in a close-loop system, which can react dynamically to unsatisfactory qualities of scheduling results.

(2) Secondly, the developed system can efficiently minimize the job tardiness or balance machine utilization rate to improve the scheduling performance using the developed heuristic rules. Four heuristics have been developed for reducing job tardiness and the facilitator can automatically select one suitable rule in order to achieve a satisfactory result efficiently. From the presented case study, it can be concluded that substantial improvement in schedule performance measure can be made.

(3) The newly generated scheduling results are obtained through re-running the optimization process of the process planning module and scheduling module, so that the function of system optimization is maximally kept and the negative effect is minimized.

8.2 Future Work

The heuristic algorithm used in the facilitator module aims at achieving multiple modification objectives, which includes modifications to the machine utilization rate and job tardiness. The heuristic rules should be further developed and extended to realize the modifications to other qualities of the scheduling result, depending on the individual manufacturing requirements.

REFERENCES

Alting and Zhang H.C., 1989, Computer aided process planning: the state-of-the-art survey. *International Journal of Production Research*, **27**(4), 553-585.

Anderson, E.J. and Nyirenda, J.C., 1990, Two new rules to minimize tardiness in a job shop, *International Journal of Production Research*, **28**(12), 2277-2292.

Artiba, A. and Elmaghraby, S. E., 1997, *The Planning and Scheduling of Production Systems: Methodologies and Applications*, London: New York: Chapman & Hall.

Baker, K.R., 1974, *Introduction to Sequencing and Scheduling*, New York: Wiley Publications.

Bedworth, D., Henderson, M.R. and Wolfe, P.M., 1991, *Computer-Integrated Design and Manufacturing*, McGraw-Hill, New York.

Chen. S.L.G. and Lin, L., 1999, Reducing total tardiness cost in manufacturing cell scheduling by a multi-factor priority rule, *International Journal of Production Research*, **37**(13), 2939-2596.

Currie, K. and Tate, A., 1991, O-plan: the open planning architecture. *Artificial Intelligence,* **52**, 29-86.

Detand, J., Kruth, J.P. and Kempenaers, J., 1992, A computer aided process planning system that increases the flexibility of manufacturing. *IPDES (Espirit project 2590) Workshop.* http://www.newcastle.research.ec.org/esp-syn/text/2590.html

Elmaraghy, H.A., Agerman, E., and Davies, B.J., 1993, Evolution and future perspective of CAPP. Annals of CIRP, **42**(2), 739-755.

Eom, D.H., Shin, H.J., Kwun, I.H., Shim, J.K., and Kim, S.S., 2002, Scheduling jobs on parallel machines with sequence-dependent family set-up times, *The International Journal of Advanced Manufacturing Technology*, **19**, 926-932.

Graves, D. and Stephen, C., 1981, A review of production scheduling. *Operations Research,* **29**(4), 646-675.

Hadavi, K., Hsu, W. L., Chen, T., and Lee, C. N., 1992, “An Architecture for Real-Time Distributed Scheduling”. *AI Magazine*, **13**, 46-56.

Huang, S. H., Zhang, H.C. and Milton L. S., 1995, A progressive approach for the integration of process planning and scheduling. *Institute of Industrial Engineers IIE Transactions,* **27**, 456-464.

James, R. J. W., 1997, Using tabu search to solve the common due date early/tardy machine scheduling problem, *Computers and Operations Research*, **24**(3), 199-208.

Kempenaers, J., Pinte, J., Detand, J., and Kruth, J.P., 1996, A collaborative process planning and scheduling system. *Advances in Engineering Software*, **25**(1), 3-8.

Khoshnevis, B. and Chen, Q., 1990, Integration of process planning and scheduling functions. *Journal of Intelligent Manufacturing*, **1**, 165-176.

Kiritsis, D., Neuendorf, K.P., and Xirouchakis, P., 1999, Petri-net model for dynamic process planning and sequence optimisation, *Advanced Engineering Software*, **30**, 375-387.

Kruth J.P. and Detand J., 1992, A CAPP system for nonlinear process plans. *Annals of CIRP*, **41**(1), 489-492.

Li, L., 2002, Agent-based computer-aided process planning for distributed manufacturing systems, *M.Eng Thesis*, National University of Singapore.

Ma, G.H., Zhang, Y.F. and Nee, A.Y.C., 2000, A simulated annealing-based optimization algorithm for process planning. *International Journal of Production Research*, **38**(12), 2671-2687.

Mamalis, A. G., Malagardis, I. and Kambouris, K., 1996, On-line integration of a process planning module with production scheduling. *International Journal of Advanced Manufacturing Technology*, **12**, 330-338.

Morton, T.E. and Pentico, D.W., 1993, *Heuristic Scheduling Systems: with applications to production systems and project management*, New York: Wiley Publications.

Muth J. F. and Thompson G. L., 1963, *Industrial Scheduling*, Englewood Cliffs: Prentice-Hall.

Palmer, G.J., 1994, An integrated approach to manufacturing planning, *Ph.D Thesis*, University of Huddersfield, 1994.

Palmer, G.J., 1996, A simulated annealing approach to integrated production scheduling. *Journal of Intelligent Manufacturing*, **7**, 163-176.

Saravanan, A. N., 2001, A Back-propagation Approach to the Integration of Process Planning and Scheduling, *M.Eng Thesis,* National University of Singapore.

Schutten, J.M.J. and Leussink, R.A.M., 1996, Parallel machine scheduling with release dates, due dates and family set-up times, *International Journal of Production Economics*, **46-47**, 119-125.

Tan, W. and Khoshnevis, B., 2000, Integration of process planning and scheduling – a review. *Journal of Intelligent Manufacturing*, **11**, 51-63.

Tapan P. B., 1999, *Multi-objective Scheduling by Genetic Algorithms*, London: Kluwer Academic Publishers.

Tonshoff H.K., Beckendroff U. and Anders N., 1989, FLEXPLAN – A concept for intelligent process planning and scheduling. *Proceedings of CIRP International workshop on Computer Aided Process Planning*, Hannover, 88-106.

Vepsalainen, A. and Morton, T.E., 1987, Priority rules and lead time estimation for job shop scheduling with weighted tardiness costs, *Management Science*, 33, 1036-1047.

Zhang, F., Zhang, Y.F., and Nee, A.Y.C., 1997, Using genetic algorithms in process planning for job shop machining. *IEEE Transactions on Evolutionary Computation*, **1**(4), 278-289.

Zhang, H.C., 1993, IPPM – a prototype to integrate process planning and job shop scheduling functions. *Annals of CIRP*, **42**(1), 513-518.

Zhang, H.C. and Mallur, S., 1994, An integrated model of process planning and production scheduling. *International Journal of Computer Integrated Manufacturing*, **7**(6), 356-364.

Zijm, W.H.M., 1995, The integration of process planning and job floor scheduling in small batch part manufacturing, *Annals of CIRP*, **44**(1), 429-432.

Zhang, F., 1997, Genetic algorithm in computer-aided process planning. *M.Eng Thesis*, National University of Singapore.